UNIVERSIDADE ESTADUAL DE CAMPINAS

Instituto de Matemática, Estatística e Computação Científica

PAULA APARECIDA KIKUCHI

Novos pré-condicionadores aplicados a problemas de programação linear e ao problema compressive sensing

Campinas

2017

Paula Aparecida Kikuchi

Novos pré-condicionadores aplicados a problemas de programação linear e ao problema compressive sensing

Tese apresentada ao Instituto de Matemática, Estatística e Computação Científica da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Doutora em Matemática Aplicada.

Orientador: Aurelio Ribeiro Leite de Oliveira

Este exemplar corresponde à versão final da Tese defendida pela aluna Paula Aparecida Kikuchi e orientada pelo Prof. Dr. Aurelio Ribeiro Leite de Oliveira.

Campinas

2017

Agência(s) de fomento e nº(s) de processo(s): CNPq, 141656/2015-8; CAPES
ORCID: http://orcid.org/0000-0002-9202-9112

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Ana Regina Machado - CRB 8/5467

K55n

Kikuchi, Paula Aparecida, 1987-
Novos pré-condicionadores aplicados a problemas de programação linear e ao problema compressive sensing / Paula Aparecida Kikuchi. – Campinas, SP : [s.n.], 2017.

Orientador: Aurelio Ribeiro Leite de Oliveira.
Tese (doutorado) – Universidade Estadual de Campinas, Instituto de Matemática, Estatística e Computação Científica.

1. Pré-condicionadores. 2. Programação linear. 3. Processamento de sinais. I. Oliveira, Aurelio Ribeiro Leite de,1962-. II. Universidade Estadual de Campinas. Instituto de Matemática, Estatística e Computação Científica. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: New preconditioners applied to linear programming and the compressive sensing problems
Palavras-chave em inglês:
Preconditioners
Linear programming
Signal processing
Área de concentração: Matemática Aplicada
Titulação: Doutora em Matemática Aplicada
Banca examinadora:
Aurelio Ribeiro Leite de Oliveira [Orientador]
Marcia Aparecida Gomes Ruggiero
João Batista Florindo
Silvana Bocanegra
Daniela Renata Cantane
Data de defesa: 07-08-2017
Programa de Pós-Graduação: Matemática Aplicada

Tese de Doutorado defendida em 07 de agosto de 2017 e aprovada pela banca examinadora composta pelos Profs. Drs.

Prof(a). Dr(a). AURELIO RIBEIRO LEITE DE OLIVEIRA

Prof(a). Dr(a). MARCIA APARECIDA GOMES RUGGIERO

Prof(a). Dr(a). JOÃO BATISTA FLORINDO

Prof(a). Dr(a). SILVANA BOCANEGRA

Prof(a). Dr(a). DANIELA RENATA CANTANE

As respectivas assinaturas dos membros encontram-se na Ata de defesa

Aos meus pais.

Agradecimentos

A Deus, por me conduzir e dar forças para enfrentar cada etapa da vida.

À minha família, e em especial aos meus pais Adelia e Paulo, pela paciência e ajuda em todos os momentos que precisei. Ter a oportunidade de estudar e receber o apoio de minha mãe é um dos maiores presentes que pude receber.

Ao meu orientador Aurelio, pela sua confiança quando aceitou me orientar no mestrado e doutorado. Por sempre ter se mostrado paciente e disposto a me ajudar.

Aos professores Márcia Ruggiero, Silvana Bocanegra, Daniela Cantane e João Batista Florindo, que compuseram a banca examinadora. As sugestões realizadas enriqueceram a versão final da tese.

Aos professores Petronio Pulino e Benjamin Bordin, que foram sempre solícitos quando tive dúvidas e os procurei.

Aos meus amigos, que suavizaram minhas dificuldades em cada passagem de fase na Unicamp. Em especial na graduação, meus amigos Karina e Vinícius. No mestrado minha amiga Julianna, a qual me ajudou e incentivou para continuar no doutorado. Por fim, no doutorado, minha amiga Daniela que foi paciente me auxiliando nas dificuldades enfrentadas durante esse período, e aos amigos Fábio, Douglas e Jorge.

Ao IMECC e à Unicamp por todas as oportunidades concedidas desde o meu ingresso em 2007. Meu desenvolvimento não só matemático, mas também como ser humano, se deve a esse meio que para mim não poderia ter sido melhor.

À CAPES - Coordenação de Aperfeiçoamento de Pessoal de Nível Superior e ao CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico, pelo apoio financeiro.

"Tudo posso Naquele que me fortalece".
(Bíblia Sagrada, Filipenses 4, 13)

Resumo

Nessa tese apresentamos novos pré-condicionadores baseados na fatoração incompleta de Cholesky e no pré-condicionador separador. Os métodos de pontos interiores são muito eficientes na solução de problemas de programação linear. Métodos iterativos são usados para resolver o sistema linear resultante quando as fatorações são densas. Como perto de uma solução os sistemas são mal condicionados, pré-condicionadores são fundamentais para seu bom desempenho. Na primeira parte desse trabalho, apresentamos resultados para problemas de pequeno e grande porte. A primeira implementação, referente a problemas de pequeno porte é feita em Matlab. O novo pré-condicionador sugerido na tese para esse caso é obtido determinando a matriz do pré-condicionador separador, e aplicando a fatoração incompleta de Cholesky nessa matriz, obtendo dessa forma o fator incompleto. O fator incompleto é o novo pré-condicionador, denominado pré-condicionador fator separador. Na segunda implementação realizada em C e Fortran, consideramos problemas de grande porte. Para obter o novo pré-condicionador, aplicamos a fatoração controlada de Cholesky na matriz definida pelo pré-condicionador separador, e utilizamos seu fator como o novo pré-condicionador. Obtivemos resultados satisfatórios em relação aos problemas de pequeno porte, e grande número de iterações do método dos gradientes conjugados para os problemas de grande porte. Na segunda parte, aplicamos um novo pré-condicionador em problemas Compressive Sensing (CS), que é uma técnica eficiente em adquirir e reconstruir sinais. Sua teoria afirma que podemos recuperar certos sinais e imagens por meio de poucas amostras. Isso é possível porque o sinal de interesse nesse trabalho é esparso em dicionários coerentes e redundantes, e as matrizes do problema satisfazem a Propriedade da Isometria Restrita (RIP). O objetivo do problema é encontrar uma solução de norma l_1 mínima, que satisfaz um sistema sobredeterminado. A não diferenciabilidade da norma l_1 é substituída pela função pseudo-Huber. Por fim, nosso problema de interesse será composto por variáveis primais e duais. Uma abordagem para resolver esse problema é o método *Primal-Dual Newton Conjugate Gradients*. Usando o fato que próximo de uma solução podemos separar as variáveis em dois grupos, as que possuem valores muito longe de zero e as com valores tendendo a zero; e que as matrizes satisfazem a RIP, mostra-se um pré-condicionador apropriado da literatura. Apresentamos um novo pré-condicionador, em que na sua construção, continuamos explorando o fato de podermos separar as variáveis em dois grupos, próximos a uma solução e as matrizes satisfazerem a RIP, como feito no método conhecido da literatura. Uma vez determinado o pré-condicionador com tais características, aplicamos a fatoração incompleta de Cholesky na matriz definida por este pré-condicionador, e usamos o fator encontrado como pré-condicionador. Os resultados obtidos foram satisfatórios em relação ao tempo e a qualidade da imagem reconstruída.

Palavras-chave: pré-condicionador. programação linear. processamento de sinais.

Abstract

In this thesis we present new preconditioners based on the incomplete Cholesky factorization and on the splitting preconditioner. Interior point methods are very efficient for solving linear programming problems. Iterative methods are used to solve the resulting linear systems when the factorizations are dense. Close to a solution these systems are ill-conditioned and preconditioning is an essential issue. In the first part of this work, we present results for small and large problems. The first implementation, referring to small problems is done in Matlab. The new preconditioner suggested in the thesis for this case is obtained by determining the matrix of the splitting preconditioner, and applying the incomplete Cholesky factorization in this matrix, thus obtaining the incomplete factor. The incomplete factor is the new preconditioner, called splitting factor preconditioner. In the second implementation performed in C and Fortran, we considered large size problems. To obtain the new preconditioner, we apply controlled Cholesky factorization to the matrix defined by the splitting preconditioner, and use its factor as the new preconditioner. We obtain satisfactory results in concerning to small problems, and large number of iterations of the conjugate gradient method for large problems. In the second part, we apply a new preconditioner in Compressive Sensing (CS) problems, which is an efficient technique to acquire and reconstruct signals. The theory about CS asserts that we can recover certain signals and images through few samples. This is possible because the signal of interest is sparse through a coherent and redundant dictionary, and the linear system matrix satisfies the Restricted Isometry Property (RIP) under reasonable assumptions. The problem consists in finding a solution with minimum 1-norm that satisfies an underdetermined linear system. The 1-norm is replaced by the pseudo-Huber function. An approach for solving this problem is the Primal-Dual Newton Conjugate Gradients method. Using the fact that, close to a solution, we can split the variables into two groups, those that have values far from zero, and those with values approaching zero and that the matrices satisfy the RIP, an appropriate preconditioner is provided in the literature. We present a new preconditioner, which in its construction, we continue to exploit the features of the problem, as previously done. Once the preconditioner exploiting these features has been computed, we apply an incomplete Cholesky factorization on it, and use the factor found as the true preconditioner. The results obtained are satisfactory in relation to the time and the quality of the reconstructed image.

Keywords: preconditioner. linear programming. signal processing.

Sumário

Introdução

Problemas de programação linear podem ser encontrados em diversos contextos, como planejamento logístico de frotas e rotas, planejamento da produção de longo, médio e curto prazo, estratégias operacionais em mineração, siderurgia, agricultura, entre outros. Na primeira parte desta tese, compreendida entre os capítulos 1 e 3, tratamos desse tipo de problema.

No Capítulo 1, a teoria de Método de Pontos Interiores é apresentada. Conceitos a respeito de otimização linear e não linear, como definições e teoremas, são abordados. Alguns Métodos de Pontos Interiores são expostos neste capítulo. Dentre as abordagens propostas na literatura para a resolução de problemas de programação linear, utilizamos o Método de Pontos interiores, em especial, utilizamos o do tipo Método de Pontos Interiores Primal-Dual Preditor-Corretor para a resolução dos problemas propostos neste trabalho.

Sabemos que a matriz do sistema linear pode não ter um bom condicionamento, assim o uso de pré-condicionadores é indispensável quando usamos métodos iterativos. No Capítulo 2 é discutido o uso de pré-condicionadores em sistemas lineares. Dissertamos sobre pré-condicionadores do tipo Fatoração Incompleta de Cholesky, em especial, do tipo Fatoração Controlada de Cholesky; em seguida falamos do pré-condicionador separador e pré-condicionadores do tipo híbrido. Por fim, apresentamos os pré-condicionadores desenvolvidos nesta tese para problemas de programação linear, que denominaremos serem do tipo Fator Separador.

O Capítulo 3 é dedicado à exposição dos resultados numéricos. O primeiro teste é realizado em Matlab, nele testamos a eficiência do novo pré-condicionador desenvolvido neste trabalho em relação ao pré-condicionador separador. No segundo teste, nosso código é incorporado ao código PCx modificado, que utiliza a linguagem C e Fortran. Terminada a aplicação para problemas de programação linear, nosso foco passa a ser nos problemas *Compressive Sensing*.

Uma técnica eficiente em adquirir e reconstruir sinais é *Compressive Sensing* (CS), também conhecida como *Compressive Sampling* (CANDÈS; WAKIN, 2008). *Compressive Sensing* é aplicado nas áreas de fotografia (HUANG et al., 2013), ressonância magnética (LUSTIG; DONOHO; PAULY, 2007), tomografia (FIROOZ; ROY, 2010), entre outras. Na segunda parte da tese, um novo método aplicado à *Compressive Sensing* é

apresentado.

No Capítulo 4, a teoria acerca de *Compressive Sensing* é exposta. Dado um sinal, sendo este esparso e as matrizes que capturam suas informações satisfazendo certas propriedades, é possível recuperarmos o sinal por meio de poucas amostras. Apresentamos a formulação de *Compressive Sensing* e, visto que trabalhar com a norma zero torna a solução do problema inviável por ser um problema combinatorial, chegamos a uma formulação com a norma l_1, dessa forma, computacionalmente tratável. Na Seção 4.2 é definida a Propriedade da Isometria Restrita (RIP), esta é utilizada para garantir que o sinal esparso seja reconstruído. Em seguida é discutido quando o sinal possui uma imagem através de dicionários, o que equivale ao sinal ser projetado em outro espaço. Por fim, explora-se as propriedades das matrizes trabalhadas, estendendo a teoria RIP para casos em que projeta-se o sinal por meio de dicionários.

No Capítulo 5, a Transformada Legendre-Fenchel é definida e obtemos, dessa maneira, uma forma alternativa de representar uma função convexa. Uma interpretação geométrica e um exemplo da transformada para a função norma l_2 são apresentados. Neste capítulo também comentamos brevemente acerca de dualidade, visto como uma forma de podermos trabalhar com um problema equivalente, mas de resolução mais fácil. A técnica de suavização de Moreau é apresentada assim como um exemplo. Por tratarmos de uma função objetivo provida da norma l_1, queremos suavizá-la podendo dessa forma obter mais informações. Reescrevendo o problema de forma a obter uma aproximação com derivadas de todas as ordens, podemos aproveitar melhor suas características. No fim da seção, estendemos o problema para o conjunto dos números complexos ($\mathbb{C}$) e reescrevemos nossa função de interesse por meio das funções Huber e Pseudo-Huber. Finalmente, é dada a formulação Primal-Dual do problema por meio da transformada Legendre Fenchel e Pseudo-Huber.

O Capítulo 6 é focado no Método de Segunda Ordem desenvolvido por Fountoulakis (2015) para o problema *Compressive Sensing*. Neste capítulo é reproduzida a descrição do método. A não diferenciabilidade da norma l_1 é tratada reescrevendo o problema por meio da função Pseudo-Huber. E por meio do conteúdo visto no Capítulo 3, chegamos a uma formulação Primal-Dual, formulação essa que será utilizada para a resolução dos problemas de interesse. Em seguida, as condições de otimalidade são determinadas. Como busca-se trabalhar no campo dos números reais, as condições de otimalidade são redefinidas. As direções primal-dual são apresentadas.

No Capítulo 7, apresentamos o nosso pré-condicionador aplicado ao problema *Compressive Sensing*. Seguindo o mesmo raciocínio feito nesta tese para problemas de programação linear, vamos aplicar a fatoração incompleta de Cholesky no pré-condicionador de Fountoulakis (2015). Tendo em vista que, como o pré-condicionador Separador, este também possui um melhor desempenho próximo à solução. Experimentos numéricos

relacionados a esse novo pré-condicionador também são apresentados. Nesse capítulo também implementamos um método considerando uma modificação no cálculo das direções primal-dual, além de um segundo método em que novas condições de otimalidade são consideradas. Por fim, apresentamos os experimentos numéricos desses novos métodos.

No Capítulo 8, apresentamos nossas conclusões do trabalho, bem como nossas perspectivas futuras.

Parte I

Programação Linear

Capítulo 1

Método de Pontos Interiores

Programação linear é utilizada para resolver um vasto campo de problemas, fundamentalmente minimizamos ou maximizamos funções lineares com restrições de igualdade ou desigualdade, também lineares. Os métodos para sua resolução são divididos em dois grupos, que são os métodos do tipo Simplex, desenvolvido inicialmente por George B. Dantzig em 1947 (BAZARAA; JARVIS; SHERALI, 2011), e os métodos de pontos interiores. O método de pontos interiores, apresentado em 1984 por Karmarkar (WRIGHT, 1997) despertou grande interesse já que, segundo ele, o método resolveria problemas de programação linear de grande porte até cinquenta vezes mais rápido que o método Simplex. Neste capítulo definimos alguns conceitos de problemas de otimização, apresentando alguns métodos de pontos interiores.

1.1 Otimização Linear

Um problema de otimização linear consiste em minimizar ou maximizar uma função objetivo linear, sujeito a um conjunto finito de restrições lineares. As restrições podem ser de igualdade ou desigualdade. A forma padrão de um problema de otimização linear é denominada problema primal e é dada por:

$$\begin{array}{ll} \text{minimizar} & c^T x \\ \text{sujeito a} & Ax = b \ , \\ & x \geqslant 0 \end{array} \tag{1.1}$$

sendo A uma matriz de restrições pertencente a $\mathbb{R}^{m\times n}$, x um vetor coluna pertencente a $\mathbb{R}^n$, cujas componentes são denominadas variáveis primais, e b e c vetores coluna pertencentes a $\mathbb{R}^m$ e $\mathbb{R}^n$, respectivamente, sendo c os custos associados aos elementos de x.

Definimos um vetor $\bar{x}$ tal que $A\bar{x} = b$, $\bar{x} \geqslant 0$, como uma solução factível, e ao conjunto de todas as soluções factíveis damos o nome de conjunto factível. Uma solução

x^* é ótima quando, além de ser uma solução factível, admite o menor valor possível para a função objetivo.

No caso das restrições serem inconsistentes, não teremos solução factível. Tal problema é chamado infactível. No caso em que existe solução factível, a função objetivo pode ser ilimitada ou limitada no domínio, sendo assim chamado de problema ilimitado e limitado, respectivamente. Não existe solução ótima se, e somente se, o problema é infactível ou ilimitado.

Para um dado problema primal, sempre podemos construir um problema associado, ao qual chamamos problema dual (BAZARAA; JARVIS; SHERALI, 2011), que consiste dos mesmos componentes dados, arranjados de uma forma diferente. O problema dual de (1.1) é dado por:

$$\begin{array}{ll} \text{maximizar} & b^T y \\ \text{sujeito a} & A^T y \leqslant c \\ & y \ livre \end{array}, \tag{1.2}$$

que é equivalente a

$$\begin{array}{ll} \text{maximizar} & b^T y \\ \text{sujeito a} & A^T y + z = c \\ & z \geqslant 0 \\ & y \ livre \end{array}, \tag{1.3}$$

sendo y um vetor coluna pertencente a $\mathbb{R}^m$, denominado vetor de variáveis duais, e z um vetor coluna pertencente a $\mathbb{R}^n$, um vetor de variáveis de folga.

A teoria da dualidade nos mostra as relações entre os problemas primal (1.1) e dual (1.2). Considerando o problema primal, seu conjunto factível e o conjunto solução fornecem muitas informações em relação ao problema dual, como também o conjunto solução e factível do problema dual proveem muitas informações acerca do problema primal. Algumas relações entre os problemas primal e dual são dadas a seguir:

Lema 1.1 (Lema Fraco da Dualidade)**.** *Se x e y são soluções factíveis para os problemas primal e dual, respectivamente, então $c^T x \geqslant b^T y$.*

Demonstração: Como a solução primal é factível, temos que $Ax = b$, assim $b^T y = x^T A^T y \leqslant x^T c = c^T x$, pois $x \geqslant 0$ e a solução dual é factível. Concluímos que a função objetivo do problema dual é um limitante inferior da função objetivo do problema primal.

Corolário 1.1. *Se $c^T x^* = b^T y^*$ e x^* e y^* são factíveis, então x^* é solução ótima do problema primal e y^* é solução ótima do problema dual.*

Demonstração: Da hipótese e do Lema Fraco da Dualidade, temos que:

$$\text{mínimo } c^T x \geqslant b^T y^* = c^T x^*.$$

Como mínimo $c^T x \leqslant c^T x^*$, chegamos que mínimo $c^T x = c^T x^*$.

Da mesma forma, da hipótese e do Lema anterior segue:

$$\text{máximo } b^T y \leqslant c^T x^* = b^T y^*.$$

Como máximo $b^T y \geqslant b^T y^*$, chegamos que máximo $b^T y = b^T y^*$.

Teorema 1.1. *Sejam x e (y, z) soluções factíveis dos problemas primal e dual, respectivamente. Uma condição necessária e suficiente para que ambas soluções sejam ótimas é que:*

Se $x_j > 0$, então $z_j = 0$.

Se $z_j > 0$, então $x_j = 0$.

Ou seja, $x_j \times z_j = 0$, para j variando de 1 a n.

Combinando os resultados acima, determinamos as Condições de Otimalidade (condições algébricas que devem ser satisfeitas pelas soluções dos problemas de programação linear).

Condições de Otimalidade

Dado um ponto (x, y, z), ele será ótimo para os problemas primal e dual se, e somente se, as seguintes condições forem satisfeitas:

$$\begin{cases} Ax = b \\ A^T y + z = c \\ XZe = 0 \\ (x, z) \geqslant 0 \end{cases}, \tag{1.4}$$

sendo X e Z matrizes diagonais formadas pelos elementos dos vetores x e z, respectivamente, e e o vetor de números um.

1.2 Otimização Não Linear

Problemas de programação não linear otimizam (maximizam ou minimizam) uma função objetivo, que satisfaz certas restrições, podendo ser estas de igualdade ou desigualdade. Note que no caso da função objetivo e as restrições serem lineares, estamos tratando de um problema de programação linear, caso a função objetivo, ou alguma restrição não for linear, nosso problema será de programação não linear. A seguir serão apresentados alguns conceitos referentes à otimização não linear.

Seja um conjunto $\Omega \subset R^n$ e sejam $f : \mathbb{R}^n \to \mathbb{R}$, $h : \mathbb{R}^n \to \mathbb{R}^m$, $h := (h_1; h_2; ...; h_m)$ e $g : \mathbb{R}^n \to \mathbb{R}^p$, $g := (g_1; g_2; ...; g_p)$, contínuas e pertencentes a C^2. Um problema de otimização não linear geral tem a seguinte forma:

$$\begin{array}{ll} \text{minimizar} & f(x) \\ \text{sujeito a} & h(x) = 0 \\ & g(x) \leqslant 0 \\ & x \in \Omega \end{array},$$

sendo $h(x) = 0$ as restrições de igualdade, $g(x) \leqslant 0$ as restrições de desigualdade e $x \in \Omega$ a restrição do conjunto Ω.

Nesse trabalho, consideramos apenas o problema com restrições de igualdade, ou seja:

$$\begin{array}{ll} \text{minimizar} & f(x) \\ \text{sujeito a} & h(x) = 0 \end{array}. \tag{1.5}$$

Chamaremos de S o conjunto formado pelos vetores $x \in \mathbb{R}^n$ tais que $h(x) = \mathbf{0}$.

Definição 1.1. *A função Lagrangeana $L : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ associada ao problema (1.5) é dada por:*

$$L(x, y) := f(x) + y^T h(x),$$

onde $y \in \mathbb{R}^m$ é o vetor dos multiplicadores de Lagrange.

Definição 1.2 (Ponto Regular)**.** *Um ponto $x^* \in S$ é dito ponto regular se $\{\nabla h_1(x^*), ..., \nabla h_m(x^*)\}$ é um conjunto linearmente independente.*

As condições de otimalidade para um problema de otimização não linear com restrição de igualdade (1.5) são dadas por:

Teorema 1.2 (Condições necessárias de primeira ordem (BAZARAA; SHERALI; SHETTY, 2013))**.** *Sejam $f, g \in C^1$ e x^* um mínimo local de f satisfazendo as restrições de igualdade. Então existe um vetor $y \in \mathbb{R}^m$ tal que*

$$\nabla f(x^*) + \sum_{i=1}^{m} y_i \nabla h_i(x^*) = \nabla f(x^*) + \nabla h(x^*)^T y = 0.$$

Tais condições podem ser obtidas derivando a função Lagrangeana com relação a x e y e igualando a zero.

$$\nabla L(x, y) = \begin{cases} \nabla f(x^*) + \nabla h(x^*)^T y = 0 \\ h(x) = \boldsymbol{0} \\ x \in \mathbb{R}^n \end{cases},$$

Teorema 1.3. *Seja x^0 um ponto regular. O subespaço tangente a este ponto é representado por:*

$$\mathcal{M}\left(x^0\right) = \left\{u \in \mathbb{R}^n \mid \nabla h\left(x^0\right)^T u = 0\right\},$$

onde $\nabla h\left(x^0\right)$ *é a matriz Jacobiana de* h *em* x^0.

Teorema 1.4 (Condições necessárias de segunda ordem)**.** *Sejam* $f, h \in C^2$ *e* x^* *um ponto regular das restrições de igualdade* $h(x) = 0$ *que é um mínimo local da função* f*. Então existe um vetor* $y \in \mathbb{R}^m$ *tal que:*

$$\nabla f(x^*) + \sum_{i=1}^{m} y_i \nabla h_i(x^*) = \boldsymbol{0},$$

e a matriz Hessiana da função Lagrangeana, $\nabla^2 L(x^*) = F(x^*) + \sum_{i=1}^{m} y_i H_i(x^*)$ *é tal que* $u^T L(x^*) u \geqslant 0$ *para todo* $u \in \mathcal{M}(x^*)$*, ou seja,* $\nabla^2 L(x^*)$ *é semi-definida positiva no conjunto considerado.*

Teorema 1.5 (Condições suficientes de segunda ordem)**.** *Sejam* $f, h \in C^2$ *e* x^* *um ponto regular do conjunto* S *tal que* $\nabla f(x^*) + y^T \nabla h(x^*) = 0$*. Suponha que* $\nabla^2 L(x^*)$ *seja definida positiva em* $\mathcal{M}$*, de modo que* x^* *satisfaz as condições necessárias de segunda ordem, então* x^* *é um mínimo local estrito de* f *em* S*.*

As demonstrações das três condições enunciadas anteriormente podem ser encontradas em Izmailov e Solodov (2009).

Convexidade

Dados os conjuntos $D \subset \mathbb{R}^n$ e $\Omega \subset \mathbb{R}^n$ tais que $D \subset \Omega$, e uma função $f : \Omega \to \mathbb{R}$. Considere o problema

$$\begin{array}{ll} \text{minimizar} & f(x) \\ \text{sujeito a} & x \in D \end{array}. \tag{1.6}$$

O conjunto D será chamado *conjunto viável* do problema, e os pontos de D serão chamados *pontos viáveis* (IZMAILOV; SOLODOV, 2009).

Definição 1.3. *(IZMAILOV; SOLODOV, 2009) Dizemos que um ponto* $\overline{x} \in D$ *é*

- *minimizador global de (1.6), se*

$$f(\overline{x}) \leqslant f(x) \qquad \forall x \in D;$$

- *minimizador local de (1.6), se existe uma vizinhança* U *de* $\overline{x}$ *tal que*

$$f(\overline{x}) \leqslant f(x) \qquad \forall x \in D \cap U.$$

Sabendo que nem sempre é fácil determinar se um mínimo local é global, apresentaremos as definições de convexidade, bem como um resultado teórico importante.

Definição 1.4. *Um subconjunto $\Omega \subset \mathbb{R}^n$ é convexo se, e somente se, para todo $x, y \in \Omega, \lambda \in [0,1]$, temos que $\lambda x + (1-\lambda)\, y \in \Omega$. Ou seja, para quaisquer dois pontos em Ω, o segmento que os une está contido em Ω.*

Definição 1.5 (Funções Convexas)**.** *Dada uma função f definida em um conjunto convexo Ω, tal função é convexa se, e somente se, para todo $x, y \in \Omega, \lambda \in [0,1]$, verifica-se*

$$f\left(\lambda x + (1-\lambda)\, y\right) \leqslant \lambda f\left(x\right) + (1-\lambda)\, f\left(y\right).$$

Se para todo $\lambda \in (0,1)$ e $x \neq y$ vale a desigualdade estrita, dizemos que f é estritamente convexa. Se $-f$ é convexa, f é côncava.

Teorema 1.6. *Seja f uma função convexa definida em um conjunto convexo $\Omega \in \mathbb{R}^n$, $f : \Omega \to \mathbb{R}$, então qualquer mínimo local é também um mínimo global de f (IZMAILOV; SOLODOV, 2009).*

1.2.1 Método de Newton Aplicado às Condições de Otimalidade para Problemas de Programação Linear

Note que podemos reescrever as condições de otimalidade (1.4) para um problema de programação linear de uma forma ligeiramente diferente, definindo uma aplicação F de $\mathbb{R}^{2n+m}$ a $\mathbb{R}^{2n+m}$:

$$F(x,y,z) = \begin{pmatrix} F_p \\ F_d \\ F_a \end{pmatrix} = \begin{pmatrix} A^T y - c + z \\ Ax - b \\ XZe \end{pmatrix} = 0,$$

supondo que $(x,z) \geqslant 0$.

Para resolver esse sistema não linear, aplicamos o Método de Newton (RUGGIERO; LOPES, 1997) às Condições de Otimalidade, aproximando F pela série de Taylor truncada e prosseguindo de forma análoga ao processo descrito em Ruggiero e Lopes (1997) referente ao Método de Newton para várias variáveis. Assim, obtemos:

$$\begin{aligned} F\left(x^{k+1}\right) &\approx F\left(x^k\right) + \nabla F\left(x^k\right)^T \left(x^{k+1} - x^k\right) = 0 \\ \Rightarrow -F\left(x^k\right) &= \nabla F\left(x^k\right)^T \left(x^{k+1} - x^k\right) \\ \Rightarrow x^{k+1} &= x^k - \left(\nabla F\left(x^k\right)\right)^{-T} F\left(x^k\right) \\ \Rightarrow x^{k+1} &= x^k + d\,, d = -\left(\nabla F\left(x^k\right)\right)^{-T} F\left(x^k\right). \end{aligned}$$

O método convergirá se certas condições forem satisfeitas. Maiores detalhes encontram-se em Ruggiero e Lopes (1997).

1.3 Método Primal-Dual

A publicação em 1984 de Karmarkar (1984) foi provavelmente o evento mais significativo em programação linear desde o método Simplex (WRIGHT, 1997). Um dos motivos do artigo ter despertado grande interesse era porque o autor afirmava que o método tinha um excelente desempenho em problemas lineares de grande porte. Esse artigo provocou uma revolução na pesquisa de problemas de programação linear, conduzindo a avanços computacionais e teóricos nessa área. A partir desse artigo e de outros trabalhos, surgiram os Métodos de Pontos Interiores, que até hoje vêm se desenvolvendo. A teoria, juntamente com experimentos computacionais, mostram que os algoritmos primal-dual possuem um desempenho melhor que outros Métodos de Pontos Interiores, assim também como possuem um desempenho melhor que o método Simplex para problemas de grande porte (WRIGHT, 1997). Descrevemos a seguir o Método Primal-Dual Afim-Escala, o Método Primal-Dual Seguidor de Caminho e o Método Preditor-Corretor.

1.3.1 Método Primal-Dual Afim-Escala

Os métodos de pontos interiores Primais-Duais encontram uma solução ótima (x^*, y^*, z^*) do problema de programação linear aplicando o Método de Newton às condições de otimalidade (desconsiderando as desigualdades: $x \geqslant 0, z \geqslant 0$) e modificando o tamanho dos passos das direções encontradas, fazendo com que sejam tais que $(x, z) > 0$ para todas as iterações. Assim, resolvemos os problemas primal e dual simultaneamente.

Iniciaremos o método com x^0, y^0 e z^0, não sendo exigido que sejam factíveis, mas apenas que $\left(x^0, z^0\right) > 0$ ou seja, que $\left(x^0, z^0\right)$ seja um ponto interior. As condições de otimalidade no ponto inicial serão dadas por $F\left(x^0, y^0, z^0\right)$, que podemos escrever como:

$$F\left(x^0, y^0, z^0\right) = \begin{pmatrix} Ax^0 - b \\ A^T y^0 + z^0 - c \\ X^0 Z^0 e \end{pmatrix} = - \begin{pmatrix} r_p^0 \\ r_d^0 \\ r_a^0 \end{pmatrix} = r\left(x_0, y_0, z_0\right),$$

onde r_p refere-se ao resíduo do problema primal, e r_d ao do dual. Agora, vamos aplicar o Método de Newton para várias variáveis às condições de otimalidade.

$$\left(x^1, y^1, z^1\right) = \left(x^0, y^0, z^0\right) - \left[J\left(x^0, y^0, z^0\right)\right]^{-1} F\left(x^0, y^0, z^0\right),$$

sendo que $J\left(x^0, y^0, z^0\right) = \begin{pmatrix} \nabla F_p^T \\ \nabla F_d^T \\ \nabla F_a^T \end{pmatrix} = \begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z^0 & 0 & X^0 \end{pmatrix}$.

Como visto na seção anterior,

$$d_0 = - \left[J\left(x^0, y^0, z^0\right)\right]^{-1} F\left(x^0, y^0, z^0\right) = \left[J\left(x^0, y^0, z^0\right)\right]^{-1} r\left(x^0, y^0, z^0\right):$$

$$d^0 = \begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z^0 & 0 & X^0 \end{pmatrix}^{-1} \begin{pmatrix} r_p^0 \\ r_d^0 \\ r_a^0 \end{pmatrix} = \begin{pmatrix} \Delta x^0 \\ \Delta y^0 \\ \Delta z^0 \end{pmatrix}.$$

Tendo em vista que resolveremos tal sistema a cada iteração, desconsideremos o índice 0. Multiplicando ambos os lados da igualdade pela matriz Jacobiana, obtemos:

$$\begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z & 0 & X \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix} = \begin{pmatrix} r_p \\ r_d \\ r_a \end{pmatrix},$$

o que nos dá o sistema:

$$A\Delta x = r_p, \tag{1.7}$$

$$A^T\Delta y + \Delta z = r_d, \tag{1.8}$$

$$Z\Delta x + X\Delta z = r_a. \tag{1.9}$$

Note que, de (1.9), obtemos $\Delta z = X^{-1}(r_a - Z\Delta x)$. Assim podemos reescrever a equação (1.8) como:

$$A^T\Delta y + X^{-1}(r_a - Z\Delta x) = r_d \Rightarrow A^T\Delta y - X^{-1}Z\Delta x = r_d - X^{-1}r_a.$$

Definindo $D = (X)^{-1}Z$, vamos substituir na equação anterior, obtendo:

$$A^T\Delta y - D\Delta x = r_d - X^{-1}r_a.$$

Logo,

$$\Delta x = D^{-1}\left(A^T\Delta y - r_d + X^{-1}r_a\right).$$

Substituindo a última equação em (1.7), chegamos a:

$$\Delta y = \left(AD^{-1}A^T\right)^{-1}\left(r_p + AD^{-1}r_d - AD^{-1}X^{-1}r_a\right).$$

Dado que a matriz A tem posto m, temos que $AD^{-1}A^T$ é definida positiva. Assim podemos calcular a decomposição de Cholesky de $AD^{-1}A^T$, por exemplo, resolvendo dois sistemas triangulares para obter Δy.

A seguir, apresentamos como é determinado o ponto inicial, quais são os critérios de convergência do método e apresentamos um resumo.

Ponto Inicial

O ponto inicial será determinado como em Mehrotra (1992). Para o problema primal:

$$\tilde{x} = A^T \left(AA^T\right)^{-1} b \Rightarrow A\tilde{x} = b,$$

$$x_i^0 = max\left\{\tilde{x}_i, \epsilon_1\right\},$$

$$\epsilon_1 = max\left\{-min\ \tilde{x}_i, \epsilon_2, \frac{\|b_1\|}{\epsilon_2 \|A\|_1}\right\},$$

$$\epsilon_2 = 100.$$

Para o problema dual:

$$y^0 = 0,$$

$$z_i^0 = \begin{cases} c_i + \epsilon_3 & se\ c_i \geqslant 0; \\ -c_i & se\ c_i \leqslant -\epsilon_3; \\ \epsilon_3 & se\ -\epsilon_3 \leqslant c_i \leqslant 0, \end{cases}$$

$$\epsilon_3 = 1 + \|c\|_1 .$$

Critério de Parada

O processo termina quando as seguintes condições são satisfeitas:

- **Factibilidade Primal:** $\dfrac{\|b - Ax\|}{\|b\| + 1} \leqslant \epsilon;$
- **Factibilidade Dual:** $\dfrac{\|c - A^T y - z\|}{\|c\| + 1} \leqslant \epsilon;$
- **Otimalidade:** $\dfrac{\left|c^T x - b^T y\right|}{1 + |c^T x| + |b^T y|} \leqslant \epsilon$ ou $\left|\dfrac{x^T z}{1 + |c^T x| + |b^T y|}\right| \leqslant \epsilon.$

Método Primal-Dual Afim-Escala

1. Ponto inicial $\left(x^0, y^0, z^0\right)$ interior, $\left(x^0, z^0\right) > 0$, sem exigência de factibilidade e $\tau \in (0, 1)$.
2. Repita até convergir:
   $r_p^k = b - Ax^k;$
   $r_d^k = c - A^T y^k - z^k;$
   $r_a^k = -X^k Z^k e;$
   $\Delta y^k = \left[A(D^k)^{-1} A^T\right]^{-1} \left[r_p^k + A(D^k)^{-1} r_d^k - A(D^k)^{-1} \left(X^k\right)^{-1} r_a^k\right];$
   $\Delta x^k = (D^k)^{-1} \left[A^T \Delta y^k - r_d^k + \left(X^k\right)^{-1} r_a^k\right];$

$$
\begin{aligned}
&\Delta z^k = \left(X^k\right)^{-1}\left[r_a^k - Z^k \Delta x^k\right];\\
&\rho_p = \min_{\Delta x_i^k<0}\left\{-\frac{x_i^k}{\Delta x_i^k}\right\};\\
&\rho_d = \min_{\Delta x_i^k<0}\left\{-\frac{z_i^k}{\Delta z_i^k}\right\};\\
&\alpha_p^k = min\left\{1, \tau\rho_p^k\right\};\\
&\alpha_d^k = min\left\{1, \tau\rho_d^k\right\};\\
&x^{k+1} = x^k + \alpha_p^k \Delta x^k \left(\alpha_p^k \text{ é tal que } x^{k+1} > 0\right);\\
&y^{k+1} = y^k + \alpha_d^k \Delta y^k;\\
&z^{k+1} = z^k + \alpha_d^k \Delta z^k \left(\alpha_d^k \text{ é tal que } z^{k+1} > 0\right).
\end{aligned}
$$

1.3.2 Método Primal-Dual Seguidor de Caminho

O método primal-dual afim escala, apesar de ter a vantagem sobre os métodos primal e dual de pontos interiores (WRIGHT, 1987) por não precisar de ponto inicial factível, não é eficiente, visto que permite que alguns produtos $x_i z_i$ aproximem-se de zero rapidamente, ou seja, aproximem-se da fronteira da região factível, fazendo com que as direções calculadas sejam distorcidas, com isso o método pode demorar a convergir ou, inclusive, não convergir.

A fim de eliminar tal problema, acrescentamos uma perturbação μ à condição de complementaridade. Assim, $XZe = \mu e$, e as novas condições de otimalidade são:

$$
\begin{cases}
Ax = b\\
A^T y + z = c\\
XZe = \mu e\\
(x, z) \geqslant 0
\end{cases},
$$

onde μ é tal que $\lim_{k\to\infty} \mu^k = 0$.

O valor estimado de μ^k, na maioria das implementações, é dado por:

$$\mu^k = \sigma^k \left(\frac{\gamma^k}{n}\right),$$

sendo $\gamma^k = \left(x^k\right)^T z^k$, e $\sigma^k \in (0, 1)$, o parâmetro de centragem.

Observe que quando $\sigma^k = 0$, temos o método afim-escala.

Quando $\sigma^k = 1 \Rightarrow \mu^k = \frac{\gamma^k}{n} \Rightarrow XZe = \frac{x^T z e}{n}$, ou seja, $x_j \times z_j$ tem o mesmo valor para todo j. A direção assim obtida é definida como *"direção de centragem"*.

Dependendo da escolha de τ e σ obtemos resultados teóricos e práticos com respeito a eficiência do método.

Agora, aplicando o Método de Newton às condições de otimalidade, obtemos o seguinte sistema linear:

$$\begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z & 0 & X \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix} = \begin{pmatrix} r_p \\ r_d \\ r_c \end{pmatrix}.$$

E prosseguimos como no método primal-dual afim-escala para encontrar as direções. As diferenças dos dois métodos são: a troca de r_a por r_c e o cálculo de μ.

Método Primal-Dual Seguidor de Caminho

1. Ponto inicial $\left(x^0, y^0, z^0\right)$ interior, $\left(x^0, z^0\right) > 0$, sem exigência de factibilidade e $\tau, \sigma \in (0,1)$.

2. Repita até convergir:
   $\mu^k = \sigma \dfrac{\gamma^k}{n}$;
   $r_p^k = b - Ax^k$;
   $r_d^k = c - A^T y^k - z^k$;
   $r_c^k = \mu^k e - X^k Z^k e$;
   $\Delta y^k = \left[A(D^k)^{-1} A^T\right]^{-1} \left[r_p^k + A(D^k)^{-1} r_d^k - A\left(Z^k\right)^{-1} r_c^k\right]$;
   $\Delta x^k = (D^k)^{-1} \left[A^T \Delta y^k - r_d^k + \left(X^k\right)^{-1} r_c^k\right]$;
   $\Delta z^k = \left(X^k\right)^{-1} \left[r_c^k - Z^k \Delta x^k\right]$;
   $\rho_p = \min\limits_{\Delta x_i^k < 0} \left\{ -\dfrac{x_i^k}{\Delta x_i^k} \right\}$;
   $\rho_d = \min\limits_{\Delta x_i^k < 0} \left\{ -\dfrac{z_i^k}{\Delta z_i^k} \right\}$;
   $\alpha_p^k = min\left\{1, \tau \rho_p^k\right\}$;
   $\alpha_d^k = min\left\{1, \tau \rho_d^k\right\}$;
   $x^{k+1} = x^k + \alpha_p^k \Delta x^k \left(\alpha_p^k \text{ é tal que } x^{k+1} > 0\right)$;
   $y^{k+1} = y^k + \alpha_d^k \Delta y^k$;
   $z^{k+1} = z^k + \alpha_d^k \Delta z^k \left(\alpha_d^k \text{ é tal que } z^{k+1} > 0\right)$.

1.3.3 Método Preditor-Corretor

O Método Preditor-Corretor de Mehrotra (MEHROTRA, 1992), que é utilizado nessa tese, é baseado em três componentes:

- Direção afim-escala, que corresponde ao passo preditor, que consiste em encontrar uma direção do problema de otimização afim-escala.

- Direção de centragem, definida pelo parâmetro σ do método primal-dual seguidor de caminho, recordando que a direção de centragem evita que as soluções ao longo das iterações se aproximem dos eixos coordenados, mantendo dessa forma, o produto $x_i z_i$ estritamente positivo. Essa direção de centragem permite-nos tomar passos maiores da direção de Newton, pois a condição de positividade somente é violada com um tamanho de passo maior do que quando consideramos μ igual a zero.
- Direção de correção, que corresponde ao passo corretor, em que calculamos a correção não linear, tentando compensar a aproximação linear do Método de Newton. Por exemplo, em um problema de programação linear na forma padrão, o termo não linear é dado por XZe.

O método preditor-corretor consiste em aplicar o Método de Newton duas vezes, utilizando a mesma Hessiana. Diferente dos métodos vistos anteriormente, que desconsideram o resíduo dos termos não lineares e sua aproximação linear pelo Método de Newton, neste método vamos introduzir as correções destas equações.

Assim, a direção de busca é obtida pela resolução de dois sistemas lineares distintos, mas com mesma matriz de coeficientes. Inicialmente, a direção *afim-escala* $(\Delta_a x, \Delta_a y, \Delta_a z)$, também chamada *direção preditora*, é calculada resolvendo o sistema:

$$\begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z & 0 & X \end{pmatrix} \begin{pmatrix} \Delta_a x \\ \Delta_a y \\ \Delta_a z \end{pmatrix} = \begin{pmatrix} r_p \\ r_d \\ r_a \end{pmatrix}, \tag{1.10}$$

onde $r_p = b - Ax$, $r_d = c - A^T y - z$ e $r_a = -XZe$. Em seguida, o lado direito é modificado fazendo $r_p = r_d = 0$, e substituindo r_a por $r_c = \mu e - \Delta_a X \Delta_a Z e$, onde o número μ é o parâmetro de centragem, $\Delta_a X = diag(\Delta_a x)$ e $\Delta_a Z = diag(\Delta_a z)$; esses seriam os resíduos da próxima iteração, se $\alpha_p = \alpha_d = 1$. O sistema obtido fica na forma:

$$\begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z & 0 & X \end{pmatrix} \begin{pmatrix} \Delta_c x \\ \Delta_c y \\ \Delta_c z \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ r_c \end{pmatrix}, \tag{1.11}$$

e com isso, obtemos a chamada direção *corretora de centragem* $(\Delta_c x, \Delta_c y, \Delta_c z)$.

A direção de busca $(\Delta x, \Delta y, \Delta z)$, por fim, será dada pela soma das duas direções anteriores:

$$(\Delta x, \Delta y, \Delta z) = (\Delta_a x, \Delta_a y, \Delta_a z) + (\Delta_c x, \Delta_c y, \Delta_c z).$$

Podemos determinar a direção de busca evitando essa soma, para isso, ao invés de resolvermos o sistema (1.11), substituímos r_a em (1.10) por r_m, em que

$$r_m = r_a + r_c = -XZe + \mu e - \Delta_a X \Delta_a Z e.$$

Assim, o sistema resolvido é dado por:

$$\begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z & 0 & X \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix} = \begin{pmatrix} r_p \\ r_d \\ r_m \end{pmatrix}. \tag{1.12}$$

De uma forma sucinta, no método preditor-corretor primeiro determinamos a direção preditora por meio do sistema (1.10) e depois resolvemos o sistema (1.12), para determinar a direção de busca. Para maiores detalhes e um entendimento melhor da teoria que compreende os Métodos de Pontos Interiores, ver Wright (1987).

A resolução de sistemas lineares, como (1.10), é o passo computacionalmente mais caro do método de pontos interiores. Felizmente, neste caso, os dois sistemas lineares a serem resolvidos compartilham da mesma matriz de coeficientes, que, em geral, é de grande porte e esparsa. Note que podemos reformular (1.10) de modo a obter sistemas lineares com matrizes que são simétricas, mais compactas e mais fáceis de manusear do que a original. Essa reformulação é possível porque em todas as iterações, as componentes (x, z) são estritamente positivas e assim as matrizes $X = diag(x)$ e $Z = diag(z)$ são inversíveis. Sendo assim, podemos eliminar a variável Δz em (1.10), obtendo o sistema equivalente:

$$\begin{pmatrix} -D & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} \Delta_a x \\ \Delta_a y \end{pmatrix} = \begin{pmatrix} r_d - (X)^{-1} r_a \\ r_p \end{pmatrix}, \tag{1.13}$$

onde $D = (X)^{-1}Z$. O sistema (1.13) é conhecido como *sistema aumentado*. Após a resolução do sistema anterior, podemos calcular Δz por meio da equação:

$$\Delta z = (X)^{-1}(r_a - Z\Delta x).$$

Como a matriz D é invertível, podemos reduzir o sistema aumentado eliminando Δx da primeira equação e substituindo na segunda, obtendo o seguinte sistema linear:

$$A(D)^{-1}A^T\Delta y = r_p + A((D)^{-1}r_d - (Z)^{-1}r_a), \tag{1.14}$$

conhecido como *equações normais*. Como $A(D)^{-1}A^T$ é o Complemento de Schur de D em

$$\begin{pmatrix} -D & A^T \\ A & 0 \end{pmatrix}, \tag{1.15}$$

também é usual dizer que usamos o complemento de Schur para resolver o sistema linear.

Nesse capítulo, discorremos sobre tipos de Métodos de Pontos Interiores, apresentando alguns exemplos. Nesse método é necessária a resolução de pelo menos um sistema linear a cada iteração. No caso desse tipo de abordagem, ao passo que o método aproxima-se de uma solução, o sistema torna-se cada vez mais mal condicionado. O capítulo a seguir apresenta uma técnica para podermos resolver sistemas com matrizes mal condicionadas de forma eficiente.

Capítulo 2

Pré-condicionadores para Sistemas Lineares

Nos Métodos de Pontos Interiores, obtém-se um sistema linear correspondente ao Método de Newton aplicado às condições de otimalidade do problema. A resolução de tal sistema pode ser feita por meio de métodos diretos, como fatoração LU e fatoração de Cholesky, ou iterativos, como o Método dos Gradientes Conjugados (MGC) (TREFETHEN; BAU III, 1997). A fatoração de Cholesky (GOLUB; Van Loan, 2012) é o método mais utilizado para a resolução do sistema linear. O cálculo dos fatores pode ser muito caro, visto que, em problemas com matrizes esparsas sua estrutura pode ser afetada com preenchimento, isto é, surgimento de elementos não nulos no fator de Cholesky, em que na matriz original os elementos correspondentes eram nulos. Dessa forma métodos iterativos mostram-se uma boa alternativa. Como a matriz do sistema é positiva definida, utilizamos para a resolução o método dos gradientes conjugados. Tal método tem um bom desempenho quando as matrizes são bem condicionadas, caso isso não ocorra, pré-condicionamento torna-se necessário. A seguir, apresentamos o que corresponde a pré-condicionar a matriz de um sistema linear, assim como algumas técnicas conhecidas.

2.1 Pré-condicionamento

Considere o seguinte sistema linear:

$$\begin{cases} x - 4y = -2 \\ 0,51x - 2y = -8 \end{cases}, \tag{2.1}$$

cuja solução é $x = -700$ e $y = 174,5$.

Agora, considere o mesmo sistema anterior, mas com uma pequena perturbação:

$$\begin{cases} x - 4y = -2 \\ 0,52x - 2y = -8 \end{cases}, \tag{2.2}$$

a solução deste último é dada por $x = 346$ e $y = 87$, ou seja, obtemos uma solução muito diferente da anterior. Isso ocorre porque o sistema é mal condicionado. O *número de condição* da matriz A ($K_p(\mathrm{A})$) é definido como (GOLUB; Van Loan, 2012):

$$K_p(\mathrm{A}) = \|\mathrm{A}\|_p \left\|\mathrm{A}^{-1}\right\|_p.$$

Temos que $K_p(\mathrm{A}) \geqslant 1$. Quanto maior o valor de $K_p(A)$, mais sensível é o sistema a perturbações, nesse caso a matriz A é dita mal condicionada. Se a matriz for simétrica definida positiva, o número de condição para a norma 2 é dado pela razão do maior e menor autovalor.

Em (GOLUB; Van Loan, 2012), o Teorema 10.2.6 define um limitante para o erro obtido no Método dos Gradientes Conjugados da seguinte forma:

Teorema 2.1. *Suponha $A \in \mathbb{R}^{n\times n}$ simétrica definida positiva e $b \in \mathbb{R}$. Sendo x_k o valor de x obtido na k-ésima iteração e $\kappa = K_2(A)$, então*

$$\|x - x_k\|_A \leqslant 2\|x - x_0\|_A \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}\right)^k,$$

sendo $\|w\|_A = \sqrt{w^T A w}$.

Do Teorema 2.1, nota-se que para $K_2(\mathrm{A}) \approx 1$, o Método dos Gradientes Conjugados comporta-se bem.

Podemos concluir então que, se queremos trabalhar com o MGC, buscamos que o sistema linear seja bem condicionado. Assim, quando o sistema linear é mal condicionado, aplicar alguma técnica a fim de obter um sistema equivalente, mas com um melhor condicionamento é fundamental, daí surge o próximo tema, pré-condicionadores.

2.1.1 Pré-condicionadores

A ideia de pré-condicionamento é conhecida há muito tempo (SAAD; Van Der Vorst, 2000), um exemplo é o trabalho proposto por Cesari (1937).

Dado o sistema $\mathrm{A}x = b$, considere o seguinte sistema equivalente:

$$M^{-1}\mathrm{A}N^{-1}\tilde{x} = \tilde{b}, \text{ onde } \quad \tilde{x} = Nx \quad \text{e} \quad \tilde{b} = M^{-1}b. \tag{2.3}$$

O sistema (2.3) é dito pré-condicionado, sendo que $M^{-1}\mathrm{A}N^{-1}$ é chamada matriz pré-condicionada.

Quando $N^T = M$, em que N^T corresponde à transposta da matriz N, se A é simétrica, obtemos um sistema pré-condicionado simétrico $M^{-1}\mathrm{A}M^{-T}$. Golub e Van Loan (2012) definem pré-condicionador como a matriz $P = M \cdot N$, assim, se $N^T = M$, o pré-condicionador P é simétrico.

Com o objetivo de alcançar uma convergência rápida, o que queremos obter é uma matriz $M^{-1}\mathrm{A}N^{-1}$ mais próxima da matriz Identidade que a matriz A, para tanto, várias técnicas de pré-condicionamento foram propostas (SAAD; Van Der Vorst, 2000; GOLUB; Van Loan, 2012). Nesta tese vamos focar em dois pré-condicionadores, a Fatoração Controlada de Cholesky (FCC) (CAMPOS; BIRKETT, 1998), e o Pré-condicionador Separador (OLIVEIRA; SORENSEN, 2005). Antes de apresentá-los formalmente, segue uma pequena abordagem dos pré-condicionadores tipo Fatoração Incompleta, cuja FCC está inclusa.

2.2 Fatorações Incompletas

Considere uma matriz A qualquer, e P uma matriz de permutação. Podemos escrever o produto PA como o produto de uma matriz triangular inferior L e superior U: PA=LU. É interessante lembrar que o resultado do produto PA é uma matriz formada pelas linhas da matriz A rearranjadas. Já o produto AP, corresponde às colunas de A rearranjadas.

Para evitar pivôs nulos, é necessária a permutação das linhas da matriz A. O processo de obtenção dos fatores L e U, é chamado Fatoração LU.

Teorema 2.2 (Teorema da Existência e Unicidade dos Fatores L e U de A (MEYER, 2000))**.** *Se $A \in \mathbb{R}^{n\times n}$ é tal que todas as submatrizes principais dominantes são não singulares, então existem e são únicos os fatores L e U tais que A=LU.*

Considerando a estratégia de pivoteamento parcial:

Teorema 2.3. *Se $A \in \mathbb{R}^{n\times n}$ é não singular, então existe uma matriz de permutação P tal que a matriz PA tem fatoração LU.*

Caso a matriz A seja simétrica positiva definida, podemos obter sua fatoração $\mathrm{PAP}^T = \mathrm{LL}^T$, conhecida como Fatoração de Cholesky, em que L é uma matriz triangular inferior com elementos da diagonal positivos. Note que a simetria de A é mantida, aplicando tanto permutações nas linhas como nas colunas de A.

Com o que vimos, podemos pensar que um bom pré-condicionador para o sistema positivo definido $\mathrm{A}x = b$, seria LL^T, pois, para A=LL^T, teremos:

$$\mathrm{L}^{-1}A\mathrm{L}^{-T} = \mathrm{L}^{-1}\mathrm{LL}^T\mathrm{L}^{-T} = I_n.$$

Mas isto recairia em um custo computacional muito caro para obter os fatores (o mesmo problema de quando opta-se por métodos diretos).

Meijerink e Van Der Vorst (1977) apresentaram a Fatoração LU incompleta mais geral. O artigo sugere que a combinação deste pré-condicionador com o Método dos Gradientes Conjugados poderia levar a uma combinação robusta e muito rápida (SAAD; Van Der Vorst, 2000). Fatorações Incompletas estão relacionadas à maioria das técnicas de pré-condicionamento que conhecemos atualmente (SAAD; Van Der Vorst, 2000).

Dada uma matriz A simétrica definida positiva esparsa, quando obtemos sua Fatoração de Cholesky, preenchimentos em entradas que são nulas em A podem ocorrer. Desta forma, L pode ser muito menos esparso que A, ocasionando a necessidade de mais espaço de armazenamento e maior custo computacional para a resolução do sistema linear.

Quando obtém-se uma Fatoração Incompleta de A, rejeita-se o preenchimento de certas entradas. Assim, podemos obter uma fatoração de $A \approx \widetilde{L}\widetilde{L}^T$, impondo que $\widetilde{L}$ apresente algum padrão de esparsidade semelhante ao de A. A prova da existência da Fatoração Incompleta de Cholesky pode ser encontrada em Meijerink e Van Der Vorst (1977).

A fim de aumentar a eficiência dos pré-condicionadores, várias estratégias foram propostas para a construção dos pré-condicionadores baseados nas Fatorações Incompletas. Para estabelecer se o elemento será descartado durante a fatoração, duas regras são estabelecidas; uma leva em consideração a posição dos elementos não nulos da matriz original e o outro leva em conta o valor numérico do preenchimento (BENZI, 2002).

A seguir destacamos uma técnica que corresponde à última regra mencionada:

- Retirada por tolerância (*drop tolerance*) - Dada uma tolerância limite τ, elementos não nulos são aceitos no fator incompleto se eles são maiores que τ.

Além disso, pode-se também fixar a quantidade de preenchimento permitido nos fatores incompletos. Tal técnica é conhecida como *fixed fill-in* e predetermina o padrão de elementos não nulos do fator incompleto, que necessariamente não é o da matriz A. Assim, podemos determinar que não haja preenchimento, ou seja, as posições no qual estão todos os elementos não nulos da matriz original coincidem com a do fator incompleto, o que equivale, por exemplo, à Fatoração de Cholesky Incompleta sem preenchimento; ou determinar que um número fixo de preenchimento seja aceito em cada coluna do fator incompleto, que é o caso da técnica fatoração de Cholesky incompleta melhorada (JONES; PLASSMANN, 1995).

Um dos problemas enfrentados nas Fatorações Incompletas, e que devemos ressaltar, são as falhas na diagonal. Sabemos do Teorema 2.2 a respeito da existência e unicidade dos Fatores LU. Caso a matriz A seja simétrica definida positiva, temos sua

Fatoração de Cholesky $A = LL^T$. Nas Fatorações Incompletas a situação é mais complicada, pois, mesmo que A admita fatoração LU (equivalentemente Fatoração de Cholesky), a Fatoração Incompleta de A pode falhar, devido à ocorrência de pivôs nulos, ou negativos, no caso da Fatoração de Cholesky.

2.2.1 Fatoração Controlada de Cholesky (FCC)

Em 1995, uma variação da Fatoração de Cholesky Incompleta foi proposta por Campos (1995), denominada Fatoração Controlada de Cholesky. Esta foi desenvolvida para resolver sistemas lineares definidos positivos, sendo inicialmente utilizada no pré-condicionamento de sistemas lineares oriundos de equações diferenciais implícitas com dependência do tempo (CAMPOS; BIRKETT, 1998).

Considere a matriz simétrica definida positiva $A \in \mathbb{R}^{n\times n}$, seu fator de Cholesky L, seu fator de Cholesky incompleto $\widetilde{L}$, e a matriz resto R:

$$A = LL^T = \widetilde{L}\widetilde{L}^T + R. \tag{2.4}$$

Note que, se considerarmos $\widetilde{L}\widetilde{L}^T$ pré-condicionador para a matriz A, obtemos:

$$\widetilde{L}^{-1}A\widetilde{L}^{-T} = \widetilde{L}^{-1}LL^T\widetilde{L}^{-T} = (\widetilde{L}^{-1}L)(\widetilde{L}^{-1}L)^T.$$

Agora, definindo $E = L - \widetilde{L}$ e substituindo na equação anterior:

$$\widetilde{L}^{-1}A\widetilde{L}^{-T} = (\widetilde{L}^{-1}(\widetilde{L}+E))(\widetilde{L}^{-1}(\widetilde{L}+E))^T =$$

$$(I_n + \widetilde{L}^{-1}E)(I_n + \widetilde{L}^{-1}E)^T.$$

É notável que, quando $\widetilde{L} \approx L$, então $E \approx 0$ e $\widetilde{L}^{-1}A\widetilde{L}^{-T} \approx I_n$.

Em Duff e Meurant (1989) mostra-se que o número de iterações do Método dos Gradientes Conjugados está diretamente relacionado com a norma de R. Da igualdade 2.4 temos que (SILVA, 2014): $R = LL^T - \widetilde{L}\widetilde{L}^T = LE^T + E\widetilde{L}^T$. Dessa forma, podemos notar que quando $\|E\| \approx 0$, então $\|R\| \approx 0$. Tendo em vista esta última implicação, o pré-condicionador FCC é construído com base na minimização da norma de Frobenius de E.

Considere o problema:

$$\min \|E\|_F^2 = \min \sum_{j=1}^{n} c_j, \quad \text{com} \quad c_j = \sum_{i=1}^{n} |l_{ij} - \widetilde{l}_{ij}|^2. \tag{2.5}$$

com l_{ij} e $\widetilde{l}_{ij}$ sendo o elemento de posição (i,j) das matrizes L e $\widetilde{L}$, respectivamente.

Agora, vamos dividir c_j da seguinte forma:

$$c_j = \sum_{k=1}^{m_j+\eta} |l_{i_kj} - \widetilde{l}_{i_kj}|^2 + \sum_{k=m_j+\eta+1}^{n} |l_{i_kj}|^2, \tag{2.6}$$

onde m_j é o número de elementos não nulos abaixo da diagonal, na j-ésima coluna da matriz A, η é o número de elementos não nulos extras permitidos por coluna e o conjunto $(i_k, 1 \leqslant k \leqslant n)$ corresponde a uma permutação das linhas $(i, 1 \leqslant i \leqslant n)$, sendo que os primeiros $(m_j + \eta)$ elementos correspondem às linhas com elementos não nulos.

Reescrevendo a equação (2.5), temos:

$$\min \sum_{j=1}^{n} c_j = \min \sum_{j=1}^{n} \left[\sum_{k=1}^{m_j+\eta} |l_{i_k j} - \widetilde{l}_{i_k j}|^2 + \sum_{k=m_j+\eta+1}^{n} |l_{i_k j}|^2 \right]. \tag{2.7}$$

Note que na última equação, os $m_j + \eta$ elementos diferentes de zero da j-ésima coluna de $\widetilde{L}$, estão no primeiro somatório. O segundo somatório contém as entradas do fator de Cholesky L, que não possuem entradas correspondentes em $\widetilde{L}$.

Considerando que $\widetilde{l}_{i_k j} \approx l_{i_k j}$, a medida que aumentamos η (ou seja, admitimos um maior preenchimento), e que $l_{i_k j}$ não é calculada, $\|E\|_F$ é minimizada com base em uma heurística que consiste em modificar o primeiro somatório em (2.6).

Com base no que foi apresentado, a seguinte estratégia é utilizada para resolver (2.5):

- Aumentar o valor do parâmetro η, permitindo dessa forma um maior preenchimento. Note que fazendo isso, c_j decrescerá, pois o primeiro somatório irá conter mais elementos. Isto é similar à técnica de *drop tolerance* (MUNKSGAARD, 1980), citada anteriormente.

- Escolher os maiores $m_j + \eta$ elementos de $\widetilde{L}$ em valor absoluto, para um η fixo. Agrupando os maiores elementos no primeiro somatório e os menores no segundo, produzimos um fator $\widetilde{L}$ ótimo para uma determinada quantidade de armazenamento.

- A matriz $\widetilde{L}$ é construída por colunas, dessa forma é necessária apenas a j-ésima coluna de A por vez, sendo armazenados os maiores elementos em valor absoluto. Assim, uma melhor aproximação é obtida com a quantidade de memória disponível.

A seguir, listamos as principais características do pré-condicionador FCC:

- **Escolha do elemento por valor**

  O padrão de esparsidade da matriz original não é considerado, ou seja, os elementos armazenados podem ou não estar nas posições correspondentes aos da matriz original.

- **Generalização da *Improved Incomplete Cholesky Factorization***

  No pré-condicionador proposto por Jones e Plassmann (1995), um número fixo de elementos não nulos em cada linha ou coluna é definido. Este número é igual ao

número de elementos não nulos da matriz original. Além disso, armazenam-se os maiores elementos em valores absolutos.

- **Incremento exponencial para evitar falhas na fatoração incompleta**

  Como foi mencionado, na fatoração incompleta o surgimento de pivôs muito pequenos e negativos (no caso da Fatoração de Cholesky Incompleta) pode vir a ocorrer. Na FCC não é diferente, visto que esta corresponde a um tipo de fatoração incompleta. A fim de evitar esse problema, a FCC descarta todo o fator $\widetilde{L}$, incrementa a diagonal de A por um fator $\sigma_i > 0$ e calcula $\widetilde{L}$ novamente. Ao invés de usar um *shift* linear σ_i a cada i-ésima tentativa de construir o novo pré-condicionador $\widetilde{L}$, como usado por Jones e Plassmann (1995), FCC usa um *shift* exponencial $\sigma_i = 5 \cdot 10^{-4} 2^{i-1}$, a fim de ter uma perturbação menor na diagonal.

- **Pré-condicionador versátil**

  Na FCC, o número de elementos nas colunas do fator incompleto $\widetilde{L}$ pode ser controlado por um parâmetro η, assim $\widetilde{L}$ poderá ter mais ou menos elementos que a matriz fatorada. O parâmetro η, pode ser definido levando-se em conta a disponibilidade de memória. A seguir apresentamos uma tabela, que relaciona η com o número de elementos de A,

Tabela 1 – Preenchimento com a FCC

η	Fator Obtido	Armazenamento
-n	$diag(A)^{-\frac{1}{2}}$	menor que nnz(A)
0	$\widetilde{L}$	igual a nnz(A)
n	L	maior que nnz(A)

em que nnz corresponde ao número de elementos não nulos.

Note que quando η =-n, obtemos o pré-condicionador escala diagonal. Quando $\eta = 0$, obtém-se um fator de Cholesky incompleto $\widetilde{L}$, que requer o mesmo espaço de armazenamento de A, sendo que o padrão de esparsidade pode não ser o mesmo. Por fim, para η =n, temos o fator de Cholesky completo L e, portanto, o armazenamento do pré-condicionador pode ser maior que o da matriz A.

Para maiores detalhes da FCC, consultar (CAMPOS, 1995; CAMPOS; BIRKETT, 1998).

2.3 Pré-Condicionador Separador

Oliveira e Sorensen (2005), propuseram o pré-condicionador separador para sistemas lineares que surgiram de métodos de pontos interiores que usam de abordagens

iterativas, em especial método dos gradientes conjugados. O pré-condicionador separador foi inicialmente desenvolvido para o sistema aumentado

$$\begin{pmatrix} -D & A^T \\ A & 0 \end{pmatrix}. \tag{2.8}$$

A construção desse pré-condicionador se baseia no comportamento da matriz D nas iterações finais do método de pontos interiores.

Considere D^k, a matriz D correspondente à $k-$ésima iteração, os elementos de D^k são dados por:

$$d_{ii}^k = \frac{z_i^k}{x_i^k}, \qquad 1 \leqslant i \leqslant n, \tag{2.9}$$

observe que todos os elementos d_{ii} são positivos.

À medida que o método converge para uma solução, podemos separar as variáveis primais e duais, x_i^k e z_i^k, em dois subconjuntos $\mathcal{B}$ e $\mathcal{N}$, tal que $\mathcal{B}$ corresponde ao subconjunto que tende a $x_i^* > 0$ e $z_i^* = 0$, e $\mathcal{N}$ ao subconjunto que tende a $x_i^* = 0$ e $z_i^* > 0$, com x^*, z^* solução do problema de programação linear. Assim temos que as variáveis irão se dividir nos subconjuntos $\mathcal{B}$ e $\mathcal{N}$,

	$\mathcal{B}$	$\mathcal{N}$
x_i^k	$\to x_i^* > 0$	$\to x_i^* = 0$
z_i^k	$\to z_i^* = 0$	$\to z_i^* > 0$

à medida que o método aproxima-se de uma solução.

Com base no que foi discutido e de (2.9), temos que $d_{ii}^k \to 0$ ou $d_{ii}^k \to \infty$, à medida que o método converge. Com isso, sistemas lineares que envolvem a matriz D tornam-se mal condicionados perto da solução. Assim, o pré-condicionador separador será dado por MM^T, em que

$$M^{-1} = \begin{pmatrix} D^{\frac{-1}{2}} & G \\ H & 0 \end{pmatrix},$$

$G = H^T D_{\mathcal{B}}^{-\frac{1}{2}} \mathcal{B}^{-1}$, $HP = [I \quad 0]$ e $AP^T = [\mathcal{B} \quad \mathcal{N}]$.

Sendo P a matriz de permutação tal que o bloco $\mathcal{B}$ possui m colunas linearmente independentes de A e $\mathcal{N}$ sendo a matriz formada pelas $n - m$ colunas restantes. A matriz pré-condicionada terá a seguinte forma:

$$M^{-1} \begin{pmatrix} -D & A^T \\ A & 0 \end{pmatrix} M^{-T} = \begin{pmatrix} -I + D^{-\frac{1}{2}} A^T G^T + GAD^{-\frac{1}{2}} & 0 \\ 0 & -D_{\mathcal{B}} \end{pmatrix}. \tag{2.10}$$

A última igualdade segue de:

$$\begin{pmatrix} D^{-\frac{1}{2}} & G \\ H & 0 \end{pmatrix} \begin{pmatrix} -D & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} D^{-\frac{1}{2}} & H^T \\ G^T & 0 \end{pmatrix} = \begin{pmatrix} -D^{-\frac{1}{2}} D + GA & D^{-\frac{1}{2}} A^T \\ -HD & HA^T \end{pmatrix} \begin{pmatrix} D^{-\frac{1}{2}} & H^T \\ G^T & 0 \end{pmatrix}$$

$$= \begin{pmatrix} -I + D^{-\frac{1}{2}}A^TG^T + GAD^{-\frac{1}{2}} & \overbrace{-D^{-\frac{1}{2}}DH^T + GAH^T}^{C} \\ \underbrace{-HDD^{-\frac{1}{2}} + HA^TG^T}_{C^T} & -HDH^T \end{pmatrix}.$$

Note que:

$$C = -D^{-\frac{1}{2}}DH^T + GAH^T = -D^{-\frac{1}{2}}DP^T \begin{pmatrix} I \\ 0 \end{pmatrix} + H^T D_{\mathcal{B}}^{-\frac{1}{2}}\mathcal{B}^{-1}AH^T$$

$$= -D^{-\frac{1}{2}} \begin{pmatrix} D_{\mathcal{B}} & D_{\mathcal{N}} \end{pmatrix} \begin{pmatrix} I \\ 0 \end{pmatrix} + H^T D_{\mathcal{B}}^{-\frac{1}{2}}\mathcal{B}^{-1}AH^T = -D^{-\frac{1}{2}} \begin{pmatrix} D_{\mathcal{B}} \\ 0 \end{pmatrix} + H^T D_{\mathcal{B}}^{-\frac{1}{2}}\mathcal{B}^{-1}AH^T.$$

Mas, pelo que vimos anteriormente $D_{\mathcal{B}} \to 0$ à medida que a solução tende ao ótimo, visto que na solução $x_i > 0$ corresponderão às colunas básicas (fazendo uma conexão ao Método Simplex), assim segue (2.10).

O pré-condicionador separador MM^T pode também ser definido para o sistema de equações normais. Considere a matriz $AD^{-1}A^T$, com $A = [\mathcal{B} \quad \mathcal{N}]P$ em que P é uma matriz de permutação tal que $\mathcal{B}$ é não singular, podemos reescrevê-la da seguinte forma:

$$AD^{-1}A^T = \mathcal{B}D_{\mathcal{B}}^{-1}\mathcal{B}^T + \mathcal{N}D_{\mathcal{N}}^{-1}\mathcal{N}^T. \tag{2.11}$$

Considerando $M = \mathcal{B}D_{\mathcal{B}}^{-\frac{1}{2}}$, ou seja, pré-multiplicando a soma obtida em (2.11), por $D_{\mathcal{B}}^{\frac{1}{2}}\mathcal{B}^{-1}$, e pós-multiplicando por seu transposto, obtemos a matriz pré-condicionada

$$D_{\mathcal{B}}^{\frac{1}{2}}\mathcal{B}^{-1}(AD^{-1}A^T)\mathcal{B}^{-T}D_{\mathcal{B}}^{\frac{1}{2}} = I + D_{\mathcal{B}}^{\frac{1}{2}}\mathcal{B}^{-1}\mathcal{N}D_{\mathcal{N}}^{-1}\mathcal{N}^T\mathcal{B}^{-T}D_{\mathcal{B}}^{\frac{1}{2}}$$

$$= I + WW^T, \text{com} \quad W = D_{\mathcal{B}}^{\frac{1}{2}}\mathcal{B}^{-1}\mathcal{N}D_{\mathcal{N}}^{-\frac{1}{2}}. \tag{2.12}$$

Note que o pré-condicionador é mais eficiente próximo a uma solução, pois (2.12) estará mais próximo da matriz identidade.

Oliveira e Sorensen (2005) propõem a matriz diagonal nas primeiras iterações, sendo o pré-condicionador separador utilizado nas últimas. Nesta abordagem, muitos problemas não convergem porque o pré-condicionador diagonal falha em muitos deles.

O preço a se pagar em resolver o sistema pré-condicionado, no lugar do sistema de equações normais (complemento de Schur), é determinar a matriz $\mathcal{B}$. O pré-condicionador separador não é uma técnica competitiva à abordagem de Método direto por Fatoração de Cholesky, sem uma implementação cuidadosa (GHIDINI; OLIVEIRA; SORENSEN, 2014).

Determinando $\mathcal{B}$

A construção da matriz bloco $\mathcal{B}$ consiste no maior custo computacional para o cálculo do pré-condicionador.

Minimizar $\|W\|$ em (2.12) é uma boa estratégia para determinar $\mathcal{B}$, pois perto da solução a matriz pré-condicionada aproxima-se da identidade, visto que para uma escolha adequada de colunas de A para formar $\mathcal{B}$, as entradas de $D_{\mathcal{B}}^{-1}$ e $D_{\mathcal{N}}$ tendem a zero. Minimizar $\|W\|$ é difícil de resolver, mas soluções aproximadas podem ser obtidas selecionando certas colunas da matriz A, assim, determinar uma forma de escolher as m colunas linearmente independentes que formam $\mathcal{B}$, de modo a obter um bom desempenho do pré-condicionador é fundamental.

Oliveira e Sorensen (2005), sugerem a escolha das primeiras m colunas linearmente independentes de AD^{-1} com mínima norma 1, ou seja, dada a matriz AD^{-1}:

- Ordenamos as colunas de AD^{-1} em ordem crescente pela norma 1.
- Determinamos as m primeiras colunas linearmente independentes por meio da fatoração LU. Essas m colunas serão as m colunas da matriz A que formarão $\mathcal{B}$.

O pré-condicionador separador pode trabalhar com o mesmo conjunto de colunas que formam $\mathcal{B}$ por várias iterações. Como consequência, o pré-condicionador calculado nessas iterações é muito barato. Note que o pré-condicionador depende, além da matriz $\mathcal{B}$, da matriz $D_{\mathcal{B}}$, assim, mesmo que a matriz $\mathcal{B}$ seja a mesma de uma iteração à outra, isso não significa que o pré-condicionador será o mesmo, visto que $D_{\mathcal{B}}$ mudará. Uma nova fatoração é realizada sempre que o Método dos Gradientes Conjugados Pré-Condicionado necessitar de muitas iterações para convergir ou o resíduo do sistema linear for maior que a tolerância permitida (GHIDINI; OLIVEIRA; SORENSEN, 2014).

Outro aspecto importante é que, se há preenchimento excessivo, a fatoração é reiniciada (GHIDINI; OLIVEIRA; SORENSEN, 2014). Assim, quando isso ocorre, a fatoração é interrompida e reordena-se as colunas linearmente independentes encontradas pelo número de entradas não nulas. A fatoração é reiniciada considerando as colunas já encontradas e o processo é repetido até m colunas linearmente independentes serem determinadas. Além disso, quando o número de entradas não nulas do fator L mais o do fator U são maiores que quatro vezes a dimensão do sistema linear, uma segunda fatoração LU, que usa técnicas sofisticadas para fatoração LU esparsa eficiente, é aplicada ao conjunto das m colunas linearmente independentes obtidas na primeira fatoração, obtendo melhorias significativas (GHIDINI; OLIVEIRA; SORENSEN, 2014).

Velazco, Oliveira e Campos (2011) ordenam as colunas de AD^{-1}, segundo a norma 2. Esta escolha mostrou melhores resultados na eficiência do pré-condicionador separador e será utilizada nas implementações desta tese.

2.4 Pré-condicionadores híbridos

Pré-condicionadores híbridos consistem na técnica em que dois ou mais pré-condicionadores distintos são utilizados a fim de tornar o método iterativo mais eficiente.

O pré-condicionador híbrido de Bocanegra, Campos e Oliveira (2007) assume a existência de duas fases durante as iterações dos Métodos de Pontos Interiores. Na primeira fase, o pré-condicionador utilizado é o construído pela Fatoração Controlada de Cholesky, e na segunda fase, o pré-condicionador utilizado é o separador, isto porque, como foi apresentado, pré-condicionadores oriundos da Fatoração Incompleta de Cholesky funcionam bem nas iterações iniciais, quando a matriz do sistema considerado ainda é bem condicionada. Já o pré-condicionador separador funciona melhor nas iterações finais.

Apesar da abordagem de Bocanegra, Campos e Oliveira (2007) apresentar um bom desempenho para várias classes de problemas de programação linear, particularmente para os que o fator de Cholesky tem um número grande de preenchimento, ela pode falhar, principalmente quando o pré-condicionador FCC perde eficiência nas iterações iniciais do método e a mudança de fase (do FCC para o separador) ocorre sem que o pré-condicionador separador esteja apto para um bom desempenho.

2.4.1 Mudança de Fase

Definir quando ocorre a mudança de fase dos pré-condicionadores é fundamental para um bom desempenho do método. Contudo, cada problema tem um comportamento diferente e, consequentemente, identificar o melhor momento para a mudança de pré-condicionadores é muito difícil.

Bocanegra, Campos e Oliveira (2007) realizam a mudança de fases nas seguintes condições:

- O *gap* inicial ($x_0^T z_0$) do problema de programação linear é reduzido por um fator de 10^6 ou
- O número de iterações internas do Método dos Gradientes Conjugados atinge $\frac{m}{2}$, onde m é a dimensão de $AD^{-1}A^T$.

A heurística também controla o valor do parâmetro η da FCC. Ou seja, determina quando preenchimentos ocorrem e a quantidade de novos elementos não nulos que devem ser inseridos em cada coluna do fator.

Na iteração inicial, o número de entradas não nulas permitidas no pré-condicionador FCC é dado da seguinte forma (BOCANEGRA; CAMPOS; OLIVEIRA, 2007):

$$\eta_0 = \begin{cases} -\dfrac{\left|AD^{-1}A^T\right|}{m}, \text{se} \quad \dfrac{\left|AD^{-1}A^T\right|}{m} > 10, \\ \\ \dfrac{\left|AD^{-1}A^T\right|}{m}, \text{caso contrário} \end{cases}, \tag{2.13}$$

em que $|\cdot|$ denota o número de elementos não nulos da matriz. Se a matriz $AD^{-1}A^T$ é densa, o parâmetro força a construção do pré-condicionador escala diagonal. Agora, se a matriz for esparsa, então um pré-condicionador com maior preenchimento é permitido.

À medida que o Método dos Gradientes Conjugados perde eficiência, o parâmetro η é incrementado. Se o número de iterações é maior que $\dfrac{m}{4}$, aumenta-se em 10 o valor de η. O processo continua até chegar no valor máximo η_{max} previamente estabelecido, ou a mudança de fase ser identificada. O valor η_{max} é baseado na quantidade de memória disponível.

Velazco, Oliveira e Campos (2011) propuseram uma outra heurística, mais eficiente e mais simples, baseada no número de iterações do Método dos Gradientes Conjugados. Nessa heurística, quando o número de iterações necessárias para convergência for maior ou igual a $\dfrac{m}{6}$, aumenta-se em 10 o valor de η. A mudança de fase ocorre quando η excede um valor fixado previamente.

2.5 Pré-condicionadores do Tipo Fator Separador

Nessa tese, novos pré-condicionadores são apresentados, eles são chamados do tipo Fator Separador. Eles se baseiam nos pré-condicionadores Fatoração Incompleta de Cholesky e Separador, apresentados anteriormente nesse capítulo. Esses pré-condicionadores são dados pelo fator obtido pela Fatoração Incompleta de Cholesky da matriz definida pelo pré-condicionador Separador. Dessa forma, se MM^T é a matriz do pré-condicionador Separador, aplicamos a Fatoração Incompleta de Cholesky nela obtendo o fator $\widehat{L}$, esse fator é o novo pré-condicionador, sendo a matriz pré-condicionada do sistema linear dada por $\widehat{L}^{-1}AD^{-1}A^T\widehat{L}^{-T}$.

Apresentamos nesse trabalho três pré-condicionadores:

Primeiro Pré-condicionador- Como foi visto, quando aplicamos o método de pontos interiores a um problema de programação linear, resolver sistemas lineares é necessário para determinar as direções. Nessa abordagem nós implementamos o método de pontos interiores preditor-corretor canalizado em Matlab. O novo pré-condicionador foi utilizado em todas as iterações (ou seja, não é uma abordagem híbrida), para o

obter aplicamos a Fatoração Incompleta de Cholesky, na matriz $\mathcal{B}D_{\mathcal{B}}^{-1}\mathcal{B}^T$ referente ao pré-condicionador Separador. O fator incompleto $\widehat{L}$ obtido, é denominado Fator Separador. Dessa forma a matriz pré-condicionada é $\widehat{L}^{-1}AD^{-1}A^T\widehat{L}^{-T}$.

Segundo Pré-condicionador- A nossa segunda abordagem corresponde a um pré-condicionador híbrido e foi implementada em C e Fortran para problemas de programação linear. Na primeira fase é utilizado o pré-condicionador Fatoração Controlada de Cholesky. Na segunda fase a estrutura do pré-condicionador Separador é calculada, isto é, $\mathcal{B}D_{\mathcal{B}}^{-1}\mathcal{B}^T$, aplica-se então a FCC nessa matriz e obtemos como no primeiro caso o pré-condicionador Fator Separador. Dessa forma na primeira fase a matriz pré-condicionada do sistema é dada por $\widetilde{L}^{-1}AD^{-1}A^T\widetilde{L}^{-T}$, em que $\widetilde{L}$ corresponde a Fatoração Controlada de Cholesky, e na segunda fase $\widehat{L}^{-1}AD^{-1}A^T\widehat{L}^{-T}$ é a matriz pré-condicionada. Note que diferente das abordagens híbridas discutidas anteriormente, na segunda fase é utilizado o pré-condicionador Fator Separador e não o pré-condicionador Separador.

Terceiro Pré-Condicionador- Nosso terceiro pré-condicionador será específico para um certo tipo de problema não linear, o *Compressive Sensing* e é implementado em Matlab. É mostrado que o pré-condicionador conhecido da literatura para esse problema assemelha-se ao pré-condicionador Separador, no sentido que possui um melhor desempenho nas iterações finais, sendo assim, aplica-se a Fatoração Incompleta de Cholesky, obtendo um Pseudo Fator Separador.

A seguir apresentamos uma tabela com as características dos três pré-condicionadores utilizados nas abordagens.

Abordagem	**Primeira**	**Segunda**	**Terceira**
Linguagem	Matlab	C e Fortran	Matlab
Tipo	PPL	PPL	*Compressive Sensing*
Fatoração	FIC	FCC	FIC
Fases	1	2	1
Matriz	$\mathcal{B}D_{\mathcal{B}}^{-1}\mathcal{B}^T$	$\mathcal{B}D_{\mathcal{B}}^{-1}\mathcal{B}^T$	$\widetilde{N}$

Na tabela, a segunda linha corresponde ao tipo de problema que estamos resolvendo, PPL indica que é um problema de programação linear, *Compressive Sensing* é o problema discutido na segunda parte desse trabalho. Na terceira linha FIC indica que foi utilizada a Fatoração Incompleta de Cholesky na matriz do pré-condicionador Separador, FCC se refere quando a fatoração utilizada foi a Fatoração Controlada de Cholesky. Na quarta linha podemos ver se o pré-condicionador utilizado é híbrido. Na última linha podemos ver as matrizes definidas pelo pré-condicionador Separador. Notamos que as duas primeiras abordagens correspondem a matriz do pré-condicionador Separador MM^T discutida anteriormente, já a terceira matriz $\widetilde{N}$ (6.18) corresponde a um tipo de pré-condicionador Separador que é definido na segunda parte da tese.

2.5.1 Abordagem Híbrida do Pré-condicionador Fator Separador

A seguir apresentamos as características do método na primeira e segunda fase, assim como a heurística adotada para a troca de fases e atualização da matriz $\mathcal{B}$.

- **Primeira Fase**

  Na primeira fase do método, a Fatoração Controlada de Cholesky é utilizada, e como feito em Velazco, Oliveira e Campos (2011), quando o número de iterações necessárias para convergência for maior ou igual a $\frac{m}{6}$, aumenta-se em 10 o valor de η. O valor de η_0 é calculado como feito em Velazco, Oliveira e Campos (2011).

- **Segunda Fase**

  Na segunda fase, o valor do parâmetro η da fase anterior é mantido.

  O valor de η será incrementado da seguinte forma:

  1. Se o número de iterações do método dos gradientes conjugados for maior ou igual que $\frac{m}{6}$, o valor de η é incrementado em 10;
  2. Se o número de iterações do método dos gradientes conjugados for maior que $\frac{m}{3}$, além do valor da factibilidade primal ser menor ou igual a 9×10^{-5}, ou seja, o método está próximo da solução, o valor de η é aumentado da seguinte forma: $\eta_k = \eta_{k-1} + 0,01 \times m$, em que m é a dimensão de $AD^{-1}A^T$ e k corresponde a k-ésima iteração. Essa alteração no valor do parâmetro η é realizada apenas uma vez, mesmo se as duas condições forem satisfeitas novamente.

  A tolerância do método dos gradientes conjugados na primeira fase é igual a 10^{-4}, e na segunda fase vale 8×10^{-6}. Essa mudança no valor da tolerância vêm do fato que analisando os experimentos numéricos é notado que o pré-condicionador Fator Separador é mais mal condicionado que o pré-condicionador Fatoração Controlada de Cholesky, dessa forma uma tolerância de menor valor é necessária. A tolerância é alterada caso a direção seja corretora e:

  - Se $m > 16000$ e a factibilidade primal for menor ou igual a 3×10^{-6}, alterando o valor da tolerância para 10^{-8};
  - Se $m \leqslant 16000$ e a factibilidade primal for menor que 10^{-5}, alterando o valor da tolerância para 10^{-8}.

  Ou seja, para problemas de maior porte é exigido que o método esteja mais próximo do ótimo para que o valor da tolerância seja alterado. A escolha desses parâmetros se deve aos experimentos numéricos realizados.

- **Troca de Fases**

  A troca de fases ocorre se:

  - O valor da factibilidade do primal for menor ou igual a 6×10^{-5}, ou;
  - Em nove iterações quaisquer, o número de iterações do método dos gradientes conjugados for maior ou igual à dimensão de $AD^{-1}A^T$.

  Obtendo um número elevado de iterações do método dos gradientes conjugados, sabemos que nosso sistema está mal condicionado, dessa forma esperamos que com a troca de fase o número de condicionamento da matriz melhore. Os valores dos parâmetros são resultados dos experimentos realizados.

- **Atualização da Matriz $\mathcal{B}$**

  Quando o método estiver na segunda fase, ou seja, o pré-condicionador Fator Separador estiver em uso, uma nova matriz $\mathcal{B}$ será calculada, quando:

  - A factibilidade primal for maior que 9×10^{-5}, ou seja, o método não estiver perto da solução e;
  - O número de iterações do método dos gradientes conjugados for maior ou igual a $\frac{m}{8}$, ou seja, a matriz pré-condicionada não está bem condicionada.

Discorrido um pouco da teoria de pré-condicionamento, e apresentados os pré-condicionadores de interesse nessa tese, no capítulo a seguir apresentamos os testes realizados.

Capítulo 3

Testes Computacionais

Neste capítulo apresentamos os testes realizados para problemas de programação linear. O primeiro teste foi realizado em Matlab, em que apresentamos os resultados obtidos utilizando o Fator Separador (oriundo da Fatoração Incompleta de Cholesky) como pré-condicionador em todas as iterações do método de pontos interiores (ou seja, não é uma abordagem híbrida), comparando com os resultados obtidos pelo método com o pré-condicionador separador, também utilizado em todas as iterações. Nesse primeiro teste nós implementamos um método de pontos interiores preditor-corretor canalizado. No nosso segundo teste, o pré-condicionador híbrido Fator Separador é integrado ao código PCx (CZYZYK et al., 1999) na sua versão modificada, proposta por Velazco, Oliveira e Campos (2010). Comparamos os resultados obtidos pelo novo método com o obtidos pelo PCx modificado (PCx_Mod).

3.1 Experimentos Computacionais em Matlab

Os experimentos numéricos foram realizados com uma implementação própria no Matlab R2014a, e realizados em um sistema operacional Microsoft Windows 10 com Intel® Core(TM) i7-5500U 2.40GHz e 8 GB de memória RAM.

Neste teste, foram considerados 32 problemas, de diferentes coleções. Eles podem ser encontrados em:

- *Computational Optimization and Applications* (COAP) <http://users.clas.ufl.edu/hager/coap/Pages/matlabpage.html> no formato mat (formato dos problemas em Matlab).

A Tabela 2 apresenta os dados dos problemas abordados. Em todos os problemas a variável primal é limitada inferiormente por zero.

Tabela 2 – Dados gerais dos problemas.

Problema	**Linhas(m)**	**Colunas(n)**	**nnz(A)**
adlittle	56	138	424
agg2	516	758	4740
agg3	516	758	4756
blend	74	114	522
czprob	929	3562	10708
degen2	444	757	4201
fit1d	24	1049	13427
fit2d	25	10524	129042
grow7	140	301	2612
grow15	300	645	5620
israel	174	316	2443
kb2	43	68	313
ken_07	2426	3602	8404
ken_11	14694	21349	49058
nug05	210	225	1050
nug06	372	486	2232
nug07	602	931	4214
nug08	912	1632	7296
qap8	912	1632	7296
pds_02	2953	7716	16571
recipe	91	204	687
sc50a	50	78	160
sc50b	50	78	148
sc105	105	163	340
sc205	205	317	665
scagr7	129	185	465
scorpion	388	466	1534
scsd1	77	760	2388
scsd6	147	1350	4316
scsd8	397	2750	8584
sctap1	300	660	1872
stocfor1	117	165	501

Nos testes, um método de pontos interiores preditor-corretor canalizado (ou seja, as variáveis podem ser limitadas) é implementado. Na resolução do sistema linear (1.3.3), vamos determinar a matriz do pré-condicionador separador ($\mathcal{B}D_{\mathcal{B}}^{-1}\mathcal{B}^T$) e depois calcular a fatoração incompleta de Cholesky dessa matriz, que corresponde ao Fator Separador da Seção 2.5, para utilizá-lo como pré-condicionador. Para realizar a fatoração incompleta de Cholesky, usamos a função ichol, do Matlab. Nela utilizamos a retirada por tolerância (*drop tolerance*), com valor igual a 10^{-3}. Também é usado *shift* na diagonal, com um valor α apropriado. Nos testes utilizamos o pré-condicionador desde a primeira iteração do método. Comparamos nossos resultados com os obtidos se utilizamos o pré-condicionador Separador em todas as iterações.

Os resultados obtidos, são apresentados nas Tabelas 3, em que o tempo obtido para a solução é dado em segundos, It corresponde ao número de iterações do método de pontos interiores, e FO indica se a solução alcançada é a ótima.

Tabela 3 – Fator Separador e Pré-condicionador Separador.

	Fator Separador			Separador		
Problema	**Tempo**	**It**	**FO**	**Tempo**	**It**	**FO**
adlittle	1,22	**22**	O	0,83	22	O
agg2	**3,54**	27	O	28,87	26	O
agg3	**3,28**	26	O	25,09	25	O
blend	0,69	**20**	O	0,26	20	O
czprob	**68,07**	59	O	144,26	55	O
degen2	3,42	21	O	-	-	NaN
fit1d	1,22	**30**	O	1,20	30	O
fit2d	88,43	**33**	O	88,40	33	O
grow7	**0,43**	**16**	O	1,01	16	O
grow15	**1,42**	**18**	O	7,26	18	O
israel	**3,85**	36	O	5,51	34	O
kb2	0,73	**19**	O	0,17	19	O
ken_07	**46,32**	**16**	O	544,31	17	O
ken_11	13782,52	24	O	-	-	-
nug05	**0,26**	**11**	O	0,79	12	O
nug06	**0,83**	**12**	O	2,72	13	O
nug07	**3,94**	**16**	O	9,68	19	O
nug08	4142,60	5303	O	3871,12	4783	O
pds_02	**446,34**	**36**	O	11505,54	316	O
qap8	211,65	307	O	30,22	15	O
recipe	6,03	**18**	O	2,07	18	O
sc50a	**0,19**	**15**	O	0,32	15	O
sc50b	**0,16**	**14**	O	0,19	14	O
sc105	**0,35**	**16**	O	0,40	16	O
sc205	2,08	**20**	O	1,25	20	O
scagr7	0,76	**23**	O	0,44	23	O
scorpion	1,56	26	O	-	-	NaN
scsd1	**0,52**	**17**	O	0,77	17	O
scsd6	**1,69**	**19**	O	2,19	19	O
scsd8	**7,82**	**18**	O	12,32	18	O
sctap1	**2,16**	**30**	O	4,85	30	O
stocfor1	0,79	**23**	O	0,50	23	O

Na coluna referente ao tempo, os valores em vermelho na Tabela 3 correspondem aos problemas em que o método com o pré-condicionador Fator Separador obteve menor tempo. Já na coluna It, que corresponde ao número de iterações necessárias para a convergência, os valores em vermelho referem-se aos problemas com menos iterações em relação ao pré-condicionador Separador. Quando o mesmo número de iterações entre os dois pré-condicionadores foi determinado, denotamos seu valor em negrito. Como o

problema ken_11 para o pré-condicionador Separador não convergiu em mais de 20 horas, na Tabela 3 indicamos isso pelo traço (−) e em azul o tempo e número de iterações obtidos pelo método com o pré-condicionador Fator Separador.

O método com o novo pré-condicionador conseguiu resolver os 32 problemas, já o com o pré-condicionador Separador conseguiu resolver 29 dos 32 aos quais os problemas degen2 e scorpion não convergiram. Indicamos em azul o tempo e o número de iterações obtidos pelo método com o pré-condicionador Fator Separador nesses dois problemas.

Em relação ao tempo, conseguimos um melhor tempo em 18 problemas, sendo que para os problemas ken_07 e pds_02 as diferenças foram expressivas. Para o problema qap8 o novo pré-condicionador não obteve um tempo favorável.

Não conseguimos diminuir o número de iterações em muitos problemas, no total foram 5. Mas podemos notar pelas partes em negrito, que o mesmo número de iterações foi obtido na maioria dos problemas, 18 no total.

3.2 Experimentos Computacionais com Problemas de Grande Porte

Os experimentos numéricos foram implementados em C, sendo a Fatoração Controlada de Cholesky implementada em Fortran, com sistema operacional Linux, distribuição Ubuntu 16.04, processador Intel ® core i7 6700, 3.4 Ghz, 32 GB de memória RAM.

Os problemas testados são de domínio público e podem ser encontrados, segundo a coleção em:

- Kennington: <http://www.netlib.org/lp/data/kennington/>
- Netlib: <http://www.netlib.org/lp/data/>
- Mészáros: <http://old.sztaki.hu/~meszaros/public_ftp/lptestset/>
- PDS: <http://www.netlib.org/lp/data/kennington/> e <http://plato.asu.edu/ftp/lptestset/pds/>
- Fome: <http://plato.asu.edu/ftp/lptestset/fome/>

O nome dos problemas, dimensão e número de elementos não nulos (nnz(A)) da matriz, assim como as coleções às quais pertencem, são apresentados na Tabela 4.

Tabela 4 – Dados gerais dos problemas.

Problema	**Linhas(m)**	**Colunas(n)**	**nnz(A)**	**Coleção**
cre-b	5.328	36382	112233	Kennington
cre-d	4094	28601	86704	Kennington
ken11	9964	16740	38157	Kennington
ken13	22365	36561	82191	Kennington
ken18	78538	128434	297886	Kennington
stocfor3	15362	22228	62960	Netlib
aa03	690	8572	60898	Mészáros
air06	690	8572	60898	Mészáros
aircraft	3754	7517	20267	Mészáros
bas1lp	5410	9824	587771	Mészáros
baxter_mat	23871	30122	106586	Mészáros
dano3mip	3201	15851	81610	Mészáros
dbic1	34205	174457	819746	Mészáros
dbir1	14025	38763	1015478	Mészáros
lpl1	34037	89383	263081	Mészáros
nsct1	15259	26430	611683	Mészáros
nsct2	15341	26512	630662	Mészáros
pcb3000	3852	7532	56422	Mészáros
pds-06	9145	28472	60075	PDS
pds-10	15637	48780	103725	PDS
pds-20	32276	106180	226494	PDS
pds-30	47957	156042	333260	PDS
pds-40	64265	214385	457538	PDS
pds-50	80328	272513	581152	PDS
pds-60	96503	332862	709178	PDS
pds-70	111885	386238	822526	PDS
pds-80	126109	430800	916852	PDS
pds-90	139741	471538	1002902	PDS
pds-100	152289	498530	1060567	PDS
fome11	11942	24286	70548	Fome
fome12	23884	48572	141096	Fome
fome13	47768	97144	282192	Fome
fome20	32276	106180	226494	Fome
fome21	64552	212360	452988	Fome

Os resultados obtidos para o método PCx_Mod e o novo método, que denotaremos por PCx_New, são apresentados na Tabelas 5 e 6. Em relação à Tabela 5, Stt indica se o método chegou à uma solução ótima (O), ou obteve o status *Unknown* (U); It corresponde ao número de iterações necessário para a convergência; T(s) é o tempo para obter a solução dado em segundos. Na Tabela 6, Trc indica em qual iteração do método há a troca de fases; Novo $\mathcal{B}$ corresponde ao número de vezes que uma fatoração foi necessária para determinar uma nova matriz $\mathcal{B}$; e por fim, o número de iterações do método dos gradientes conjugados, realizados na segunda fase do método, é mostrado na coluna pcg.

Tabela 5 – Resultados para PCx_Mod e PCx_New.

	PCx_Mod			PCx_New		
Problema	**Stt**	**It**	**T(s)**	**Stt**	**It**	**T(s)**
cre-b	O	43	34,55	O	44	45,74
cre-d	O	42	21,93	O	43	22,85
ken11	O	22	4,26	O	**22**	**2,60**
ken13	O	30	92,26	O	34	240,76
ken18	O	40	966,95	O	50	10710,38
stocfor3	O	32	688,99	O	33	955,43
aa03	O	22	4,98	O	**22**	9,92
air06	O	22	4,95	O	**22**	10,19
aircraft	O	24	10,05	U	-	-
bas1lp	U	-	-	O	**16**	**273,82**
baxter_mat	O	35	499,62	U	-	-
dano3mip	U	-	-	U	-	-
dbic1	O	55	652,25	U	-	-
dbir1	U	-	-	U	-	-
lpl1	O	73	1587,98	O	**73**	2503,89
nsct1	O	30	907,46	O	**30**	1348,29
nsct2	O	36	735,0	O	37	1923,95
pcb3000	O	29	38,31	O	**29**	41,96
pds-06	O	39	10,93	O	**39**	**10,63**
pds-10	O	46	28,56	O	47	**28,54**
pds-20	O	61	173,06	O	62	**170,26**
pds-30	O	73	278,63	O	**73**	**257,58**
pds-40	O	77	437,70	O	78	460,13
pds-50	O	80	1016,75	O	**80**	1165,12
pds-60	O	82	1571,73	O	**81**	2048,64
pds-70	O	85	1569,88	O	**84**	1983,50
pds-80	O	83	1795,26	O	**82**	2294,17
pds-90	O	82	2564,83	O	**81**	2956,93
pds-100	O	87	3757,32	O	**86**	4135,79
fome11	U	-	-	O	**50**	**548,03**
fome12	U	-	-	O	**48**	**1466,05**
fome13	U	-	-	O	**50**	**6857,07**
fome20	O	61	172,30	O	62	**170,52**
fome21	O	73	774,42	O	**73**	**710,31**

Tabela 6 – Resultados para PCx_Mod e PCx_New.

	PCx_Mod			PCx_New		
Problema	**Trc**	**Novo $\mathcal{B}$**	**pcg**	**Trc**	**Novo $\mathcal{B}$**	**pcg**
cre-b	40	1	670	40	1	20253
cre-d	40	1	204	40	1	7015
ken11	21	1	15	21	1	337
ken13	26	1	203	26	1	163579
ken18	33	1	1628	31	1	1824174
stocfor3	31	1	2	31	1	52780
aa03	10	3	1792	10	11	10874
air06	10	3	1792	10	11	10874
aircraft	18	1	598	18	-	-
bas1lp	11	-	-	11	4	37330
baxter_mat	34	1	148	34	-	-
dano3mip	19	-	-	19	-	-
dbic1	53	1	2219	0	-	-
dbir1	36	-	-	0	-	-
lpl1	66	2	24781	66	1	167220
nsct1	10	5	45167	10	18	206163
nsct2	10	5	50357	10	26	364625
pcb3000	22	3	3822	22	1	40309
pds-06	29	2	7583	30	1	**4457**
pds-10	35	2	9986	34	1	**8090**
pds-20	52	2	9348	50	1	18391
pds-30	65	1	17682	65	1	17828
pds-40	69	1	14694	70	1	27366
pds-50	73	1	21909	73	1	52333
pds-60	74	1	41257	73	1	103463
pds-70	78	1	13188	79	1	56461
pds-80	76	1	16081	76	1	73862
pds-90	76	1	25426	76	1	62866
pds-100	80	1	65197	80	1	107260
fome11	43	-	-	43	1	91869
fome12	42	-	-	42	1	99966
fome13	43	-	-	43	1	159307
fome20	52	2	9348	50	1	18391
fome21	61	2	34759	60	1	60642

Na coluna referente ao tempo, os valores em vermelho na Tabela 5, correspondem aos problemas em que o método PCx_New obteve o menor tempo. Na coluna correspondente ao número de iterações do método de pontos interiores, os valores em vermelho referem-se aos problemas que é alcançado menos iterações, em relação ao método PCx_Mod. Quando o mesmo número de iterações entre os dois métodos foi determinado, denotamos seu valor em negrito. As partes em azul correspondem ao número de iterações do método de pontos interiores e tempo obtidos pelo método PCx_New nos problemas em

que o método PCx_Mod obteve o status *Unknown*. Na coluna correspondente ao número de iterações do método dos gradientes conjugados, contida na Tabela 6, destacamos em vermelho os problemas que PC_New obteve o menor número de iterações.

Dos 34 problemas testados, o novo método resolveu 29 e PCx_Mod 28. Na maioria dos problemas PCx_Mod obteve o menor tempo; destacamos que para ken_18 o tempo obtido pelo PCx_New foi maior que dez vezes o obtido pelo PCx_Mod, sendo que para ken_13 e nsct2 o tempo foi maior que duas vezes.

Podemos associar o maior tempo e maior número de iterações do método de pontos interiores com o número de iterações realizadas no método dos gradientes conjugados. Apenas para os problemas pds-06 e pds-10 conseguimos um número de iterações de pcg menor. Note que para o problema stocfor3, apesar de obtermos um número de iterações aproximadamente 26000 vezes maior, a diferença de tempos não foi extremamente grande, visto que a troca de fases ocorre em uma iteração próxima à final. Já para o problema ken_18, apesar do número de iterações do método dos gradientes conjugados obtidos para o método PCx_New ser aproximadamente 1100 vezes maior, o tempo obtido é bem mais significativo, devido ao maior número de iterações do método realizadas na segunda fase. Obtivemos alguns resultados semelhantes em relação ao tempo nos dois métodos, o que pode ser visto por exemplo, nos problemas iniciais da coleção PDS, fome20 e fome21, pcb3000, cre-d (que obteve o número de iterações de pcg aproximadamente 34 vezes maior que PCx_Mod), apesar de alcançarmos um número bem maior de iterações no método dos gradientes conjugados.

Na família de problemas FOME, conseguimos resolver todos os problemas, alcançando os melhores resultados comparado ao PCx_Mod.

Note que para os problemas dbc1 e dbir1, da Coleção Mészáros, o método PCx_New não trocou de fase e também não convergiu.

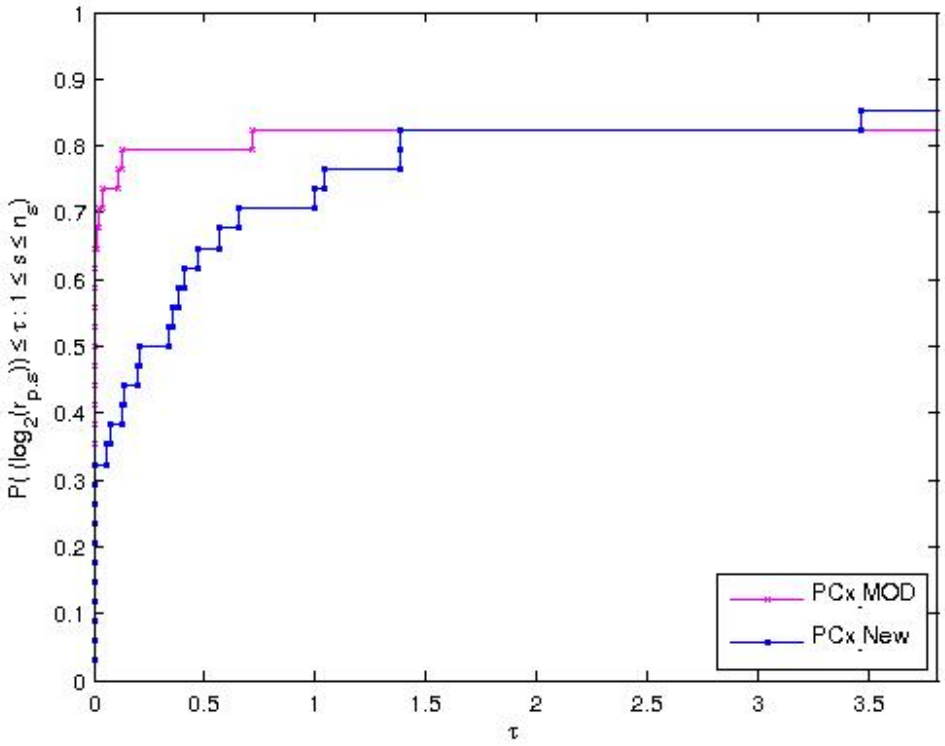

Figura 1 – Perfil de desempenho em relação ao tempo.

A Figura 1 mostra o perfil de desempenho (DOLAN; MORÉ, 2002) em relação ao tempo. Por meio dela podemos dizer que o método PCx_Mod é mais eficiente, resolvendo cerca de 64% dos problemas teste em tempo total reduzido, enquanto o método com o pré-condicionador Fator Separador resolve aproximadamente 33%. Quanto à robustez, temos que PCx_New resolve mais problemas.

Nesse capítulo, apresentamos os resultados obtidos para problemas de programação linear. Os capítulos seguintes tratam do problema *Compressive Sensing*, em que desenvolve-se um novo pré-condicionador para esse tipo de problema. Esse pré-condicionador segue a mesma ideia do que foi apresentado para problemas de programação linear. É aplicada a fatoração incompleta de Cholesky em um pré-condicionador que possui melhor desempenho nas iterações finais.

Parte II

Compressive Sensing

Capítulo 4

Compressive Sensing(CS)

Neste capítulo, apresentamos a teoria acerca de *Compressive Sensing* (CS), esta afirma que pode-se recuperar certos sinais e imagens através de poucas amostras ou medidas, comparado aos métodos tradicionais (CANDÈS; WAKIN, 2008). CS torna isso possível pois visa duas propriedades: esparsidade do sinal de interesse e a matriz a ser trabalhada obedecer a condição da Propriedade da Isometria Restrita (*Restricted Isometry Property*) (CANDÈS; TAO, 2005). Temos que:

- Esparsidade - Explora o fato de que muitos sinais naturais são esparsos em bases apropriadas.
- Propriedade da Isometria Restrita (RIP) - Mostra a eficiência das matrizes que capturam as informações do sinal esparso.

Para mais detalhes ver Duarte e Baraniuk (2013).

O objetivo é criar um novo Método de Segunda Ordem aplicado à *Compressive Sensing*. Na tese de Fountoulakis (2015) é apresentado um método denominado *Primal-Dual Newton Conjugate Gradients* (pdNCG). Vamos modificar tal método, buscando resultados mais eficientes.

De uma forma geral, nosso objetivo é resolver um problema da seguinte forma:

$$\min f_\tau(x) := \tau\psi(x) + \varphi(x), \tag{4.1}$$

onde $x \in \mathbb{R}^n$ e τ é um parâmetro positivo. Sendo $\psi : \mathbb{R}^n \to \mathbb{R}$ uma função convexa possivelmente não suave e $\varphi : \mathbb{R}^n \to \mathbb{R}$ uma função convexa. Para a resolução desse problema, é utilizado um método em que, em toda iteração, uma função convexa $Q(y; x)$ é criada e resolvemos o subproblema

$$\min_y Q(y; x). \tag{4.2}$$

Essa função aproxima-se localmente da função f_τ no ponto x.

Métodos de Primeira Ordem - Uma aproximação Q para o Problema (4.1) é definida como:

$$Q(y;x) = \tau\psi(y) + \varphi(x) + \nabla\varphi(x)^T(y-x) + \frac{1}{2}(y-x)^T H(x)(y-x), \tag{4.3}$$

onde $H(x) \in \mathbb{R}^{n\times n}$ é uma matriz positiva definida para todo x. Agora nos resta definir $H(x)$. Idealmente, ela deveria ser similar a $\nabla^2\varphi(x)$ para poder preservar as características da função φ no ponto x. Contudo, isto faz o Subproblema (4.2) tão difícil a ser minimizado quanto o problema original (4.1). Para a resolução do problema, ao invés de termos uma matriz H, em que as informações da curvatura da função φ são bem representadas, vamos ter uma matriz que não é uma boa aproximação, mas oferece soluções com baixo custo computacional para o Subproblema (4.2). Embora quase todas as propriedades de $\nabla^2\varphi(x)$ sejam perdidas, a complexidade por iteração é baixa, o que compensa em alguns problemas.

Os métodos que usam aproximações simples de $\nabla^2\varphi(x)$, são chamados métodos de primeira ordem. Eles são eficientes para problemas de grande porte bem condicionados, da forma do Problema (4.1).

Métodos de Segunda Ordem - Nesse segundo método, a função não suave ψ é aproximada a uma função suave ψ_μ. Como veremos, se $\psi(x) = \|x\|_1$, então ψ_μ pode ser a função pseudo-Huber, que explicaremos com mais detalhes no Capítulo 3. Usando a função suave ψ_μ, o Problema (4.1) é reescrito como:

$$\min f_\tau^\mu(x) := \tau\psi_\mu(x) + \varphi(x). \tag{4.4}$$

O Problema (4.4) tem a melhor aproximação ao Problema 4.1, para o menor μ. Note que f_τ^μ é uma função suave, a qual possui derivadas de todas as ordens. Dessa forma, teremos uso das informações de segunda ordem da função f_τ e as informações das curvaturas poderão ser exploradas. Para o Problema 4.4, a aproximação convexa Q no ponto x é dada por:

$$Q(y;x) = f_\tau^\mu(x) + \nabla f_\tau^\mu(x)^T(y-x) + \frac{1}{2}(y-x)^T H(x)(y-x), \tag{4.5}$$

onde $H(x)$ é uma matriz positiva definida, que satisfaz $H(x) \approx \nabla^2 f_\tau^\mu(x)$. Para $H(x) = \nabla^2 f_\tau^\mu(x)$, Q aproxima-se melhor de f_τ^μ em x. Porém, minimizar o Subproblema 4.2 pode ser uma operação com custo computacional elevado. Para suprir essa dificuldade, contamos com uma solução aproximada de (4.2) usando algum método iterativo que requeira apenas operações de produto matriz-vetor com a matriz $\nabla^2 f_\tau^\mu(x)$. Tais métodos são aproximadamente de segunda ordem, sendo eficientes quando uma precisão maior é requerida. Frequentemente afirma-se que métodos de segunda ordem não se adaptam com a dimensão do problema, por causa do maior custo em resolver aproximadamente os subproblemas em (4.2). Isso é baseado no fato que assume-se que todas as informações de segunda ordem são usadas para resolver os subproblemas. Mas claramente isso não é necessário, pois aproximações de informações de segunda ordem já são suficientes.

É provado em Fountoulakis (2015) que para problemas mal condicionados não triviais métodos de segunda ordem são eficientes.

4.1 Formulação de *Compressive Sensing*

CS busca a solução do sistema sobre-determinado

$$Ax = \hat{b}, \tag{4.6}$$

com $A \in \mathbb{R}^{m\times n}$, $x \in \mathbb{R}^n$, $\hat{b} \in \mathbb{R}^m$. Como foi mencionado, nosso interesse está em obter um vetor x esparso, ou seja, com maior número possível de entradas iguais a zero; denotamos a solução mais esparsa por $\hat{x}$.

Como queremos a solução mais esparsa que satisfaça (4.6), podemos escrever o problema como:

$$\begin{array}{ll} \text{minimizar} & \|x\|_0 \\ \text{sujeito a} & Ax = \hat{b} \end{array}, \tag{4.7}$$

onde $\|\cdot\|_0$ é a norma zero (retorna o número de elementos diferentes de zero do vetor). Ressaltamos o abuso de linguagem ao denominar a função $\|\cdot\|_0$ de norma, tendo em vista que não satisfaz todas as condições da definição de norma.

Na prática não trabalhamos com este problema, já que o uso da norma zero faz dele um problema combinatorial (WIPF; RAO, 2005). Mostra-se que em certas situações (CANDÈS, 2006) a recuperação exata da solução esparsa $\hat{x}$ de (4.6) pode ser encontrada com alta probabilidade resolvendo o seguinte problema chamado de *Basis Pursuit* (KIKUCHI, 2013; CHEN; DONOHO; SAUNDERS, 2001):

$$\begin{array}{ll} \text{minimizar} & \|x\|_1 \\ \text{sujeito a} & Ax = \hat{b} \end{array}. \tag{4.8}$$

A função objetivo de (4.8), ao contrário da tratada em (4.7) com a norma zero, pode ser reformulada como uma função linear com restrições não negativas, sendo assim, reescrito como um problema de programação linear e computacionalmente tratável.

Dada uma reformulação linear de (4.8), métodos de otimização eficientes, tais como os Métodos de Pontos Interiores, podem ser usados para determinar a solução $\hat{x}$ esparsa de (4.8).

Em aplicações reais o vetor $\hat{b}$ de (4.8) é frequentemente corrompido por ruído, dessa forma temos $b = \hat{b} + e$ e (4.6) torna-se:

$$Ax = b = \hat{b} + e, \tag{4.9}$$

onde $e \in \mathbb{R}^m$ denota o erro. Nesse caso, a solução de:

$$\begin{array}{ll} \text{minimizar} & \|x\|_1 \\ \text{sujeito a} & Ax = b \end{array}, \tag{4.10}$$

pode ser encontrada através de *Basis Pursuit Denoising* (BPDN), que pode ser encontrado em Kikuchi (2013) e Chen, Donoho e Saunders (2001). Assim, resolvemos o problema:

$$\text{minimizar} \quad \tau \|x\|_1 + \frac{1}{2} \|Ax - b\|_2^2 \,, \tag{4.11}$$

onde $\tau \geqslant 0$, ou um dos seguinte problemas:

$$\begin{array}{ll} \text{minimizar} & \|Ax - b\|_2 \\ \text{sujeito a} & \|x\|_1 \leqslant \epsilon_1 \end{array} \,, \tag{4.12}$$

ou

$$\begin{array}{ll} \text{minimizar} & \|x\|_1 \\ \text{sujeito a} & \|Ax - b\|_2 \leqslant \epsilon_2 \end{array} \,, \tag{4.13}$$

onde ϵ_1 e ϵ_2 são escalares positivos que regulam a esparsidade e o limitante superior do erro do ruído, respectivamente.

É provado em Rockafellar (2015) que o problema (4.11) é equivalente aos problemas (4.12) e (4.13) para valores específicos dos escalares τ, ϵ_1 e ϵ_2.

4.2 Matrizes de CS e suas Propriedades

Para garantir que o sinal esparso x de interesse seja reconstruído, ou seja, garantir a recuperação do vetor esparso de (4.6) por meio da minimização da norma l_1 (4.8), é definida a Propriedade da Isometria Restrita (RIP) (CANDÈS, 2008). Segue a definição:

Definição 4.1. *A Constante da Isometria Restrita δ_q, sendo $\delta_q \geqslant 0$, de uma matriz $A \in \mathbb{R}^{m \times n}$ é definida como o menor δ_q tal que*

$$(1 - \delta_q) \|x\|_2^2 \leqslant \|Ax\|_2^2 \leqslant (1 + \delta_q) \|x\|_2^2 \tag{4.14}$$

para todos os vetores q-esparsos $x \in \mathbb{R}^n$, sendo que um vetor q-esparso é tal que possui no máximo q elementos diferentes de zero.

A matriz que satisfaz a desigualdade acima, para um dado δ_q, é dita satisfazer a RIP com constante de isometria restrita δ_q.

Equivalentemente, este valor é dado por:

$$\delta_q = \text{maximizar} \left\| A_q^T A_q - I \right\|_{2 \to 2} \,,$$

onde A_q corresponde a submatriz de A formada por no máximo q colunas, e

$$\|A\|_{p \to t} := \sup_{\|x\|_p \leqslant 1} \|Ax\|_t = \sup_{\|x\|_p = 1} \|Ax\|_t \,.$$

A demonstração da equivalência pode ser encontrada no Capítulo 6 de Foucart e Rauhut (2013).

Note que, para valores pequenos de δ_q, a definição requer que todas as colunas das submatrizes de A com no máximo q colunas sejam bem condicionadas. Se tivéssemos as matrizes A e x, sendo A mal condicionada, teríamos por exemplo:

$$A = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \qquad \text{e} \qquad x = \begin{bmatrix} a \\ b \end{bmatrix},$$

multiplicando a matriz e o vetor obteríamos:

$$Ax = \begin{bmatrix} a \\ a \end{bmatrix},$$

o que implica que as normas de $\|Ax\|_2^2$ e $\|x\|_2^2$ não necessariamente têm o valor próximo (Ax depende apenas de a); e temos que a desigualdade (4.14) deve ser válida para todos os vetores q-esparsos.

Lembrando também que Operadores Ortogonais Q são invariantes pela norma l_2, ou seja, $\|Qy\|_2 = \|y\|_2$, se tivéssemos $\delta_q = 0$ segue da desigualdade (4.14) que $\|Ax\|_2 = \|x\|_2$, podendo concluir que δ_q mede o quão perto as colunas de A comportam-se como um sistema ortonormal (mas apenas quando restringimos as combinações esparsas envolvendo não mais que q vetores) (CANDÈS; TAO, 2005).

Concentrando no caso onde x é esparso, gostaríamos de encontrar a solução de (4.7). Como já foi mencionado, isso é um problema combinatorial difícil. Contudo, o Teorema 1.1 de Candès (2008) garante que os problemas l_0 e l_1 são de fato formalmente equivalentes no seguinte sentido:

- Se $\delta_{2q} < 1$, o problema l_0 tem uma única solução q-esparsa;
- Se $\delta_{2q} < \sqrt{2} - 1$, a solução para o problema l_1 é aquela do problema l_0. Em outras palavras, a relaxação convexa é exata.

De fato, se $\delta_{2q} < 1$, qualquer solução q-esparsa é única (CANDÈS; TAO, 2005). Em outra direção, suponha $\delta_{2q} = 1$. Então $2q$ colunas de A podem ser LD (linearmente dependentes), em que existe um vetor h $2q$-esparso obedecendo $Ah = 0$. Podemos então decompor h como $x - x'$, onde x e x' são q-esparsos, isto nos dá $Ax = Ax'$, como podemos ver a seguir:

$$(1 - \delta_q) \|h\|_2^2 \leqslant \|Ah\|_2^2 \leqslant (1 + \delta_q) \|h\|_2^2,$$

com $\delta_{2q} = 1$:

$$0 \leqslant \|Ah\|_2^2 \leqslant (2) \|h\|_2^2,$$

Se $Ah = 0$ e $h = x - x'$,

$$\Rightarrow Ax = Ax'.$$

O que implica que não podemos reconstruir todo vetor q-esparso por qualquer método.

Foucart (2010) demonstra o seguinte teorema que estabelece a relação entre a Propriedade da Isometria Restrita (RIP) e a recuperação do sinal (vetor) esparso.

Teorema 4.1. *Todo vetor q-esparso $x \in \mathbb{R}^n$ satisfazendo $Ax = \hat{b}$ é a única solução de (4.8) se*

$$\delta_{2q} < \frac{3}{4 + \sqrt{6}} \approx 0,4652.$$

A RIP também implica recuperação estável via minimização da norma l_1 para vetores que podem ser bem aproximados a vetores esparsos.

Podemos notar que RIP depende das dimensões da matriz A. Se n é grande, isso implica maior dificuldade para recuperar o vetor x, portanto teremos um δ_q maior. Por outro lado, m é a dimensão do vetor b que possuímos, e assim quanto menor a dimensão de b, menor será o δ_q, assim teremos uma melhor recuperação do vetor de interesse.

Em aplicações é frequente um sinal ser esparso com respeito a uma base diferente das que as medidas b foram tomadas. Dado um vetor z, vamos assumir que z é esparso com respeito a base formada pelas colunas de uma matriz unitária Ψ (matriz de esparsidade), ou seja, $z = \Psi x$ para um vetor q-esparso x. Adicionalmente, assuma que o vetor z é escrito como combinação linear de elementos de uma base formada por colunas de uma matriz unitária Φ (matriz de medição), e aplica-se R_m no vetor de coeficientes desta combinação, ou seja : $y = R_m \Phi^T z$, onde R_m é um operador de amostragem aleatória que satisfaz $R_m R_m^T = I_m$, onde I_m é a matriz identidade de ordem m. Dessa forma, temos:

$$z = \Psi x \qquad \text{e} \qquad y = R_m \phi^T z \qquad \Rightarrow y = R_m \Phi^T \Psi x$$

com x q-esparso e $y \in \mathbb{R}^m$. Assim, a matriz A em (4.6) é igual a $R_m \Phi^T \Psi$ e suas linhas são ortonormais:

$$AA^T = I_m. \tag{4.15}$$

A propriedade de recuperação da matriz A depende do valor $\mu(\Phi, \Psi)$, chamado coerência mútua dos sistemas de esparsidade (Ψ) e medidas (Φ) (DONOHO; HUO, 2001). A coerência mútua é definida da seguinte forma:

$$\mu(\Phi, \Psi) = \max_{i,j} \left| \Phi_i^T \Psi_j \right|,$$

onde Φ_i, Ψ_i correspondem à i-ésima coluna da matriz Φ e Ψ, respectivamente.

É mostrado em Elad e Bruckstein (2002), Donoho e Huo (2001) que para qualquer par de bases ortonormais Φ_1, Φ_2 de $\mathbb{R}^N$,

$$\frac{1}{\sqrt{N}} \leqslant \mu(\Phi_1, \Phi_2) \leqslant 1, \tag{4.16}$$

sendo que $\mu(\Phi_1, \Phi_2) = 1$ quando Φ_1 e Φ_2 tem um vetor coluna em comum.

Note que o resultado (4.16) é intuitivo, pois como $\Phi^T\Psi$ é ortogonal, e quando buscamos $\max_{i,j} \left|\Phi_i^T \Psi_j\right|$, estamos buscando o máximo valor em módulo entre todos os coeficientes do produto $\Phi^T\Psi$, assim, sendo $\Theta = \Phi^T\Psi$ ortogonal:

$$\sum_{j=1}^{N} \Theta_{i,j}^2 = 1.$$

O valor médio das entradas dos vetores coluna, ou linha, é dessa forma $\frac{1}{\sqrt{N}}$, o que implica que o menor valor máximo das entradas do vetor é $\frac{1}{\sqrt{N}}$. Logo, o máximo valor se dá quando uma entrada for 1 e as demais nulas.

Em alguns trabalhos (CANDÈS; ROMBERG, 2007), considera-se a coerência mútua ($\mu(\Phi, \Psi)$) sendo limitada por diferentes valores dos dados em (4.16), nessa caso temos $1 \leqslant \mu \leqslant \sqrt{N}$. Isso ocorre porque nesses trabalhos considera-se que a matriz ortogonal satisfaz $\Theta^T\Theta = NI$, diferentemente da definição formal de matriz ortogonal (GOLUB; Van Loan, 2012), sendo tal que $\Theta^T\Theta = I$. Nesses trabalhos (CANDÈS; ROMBERG, 2007) a matriz ortogonal formal é definida como matriz ortonormal.

Resumidamente, em uma definição trabalhamos com matrizes ortogonais usuais, e na outra com a matriz ortogonal tal que $\Theta^T\Theta = NI$.

Convencionamos usar a definição formal de matriz ortogonal ($\Theta^T\Theta = I$) nessa tese.

Em uma análise mais geométrica, pela definição de $\mu(\Phi, \Psi)$, que envolve um produto interno (portanto o ângulo entre os vetores colunas das matrizes Φ e Ψ em questão), notamos que quanto mais os vetores Φ_i e Ψ_i estão separados (em relação as suas inclinações), menor o valor de $\mu(\Phi, \Psi)$. Como calculamos o valor máximo do produto interno entre as colunas das duas matrizes, procuramos dentre todas a menor distância. Se temos os vetores Φ_i e Ψ_i próximos um do outro, podemos suspeitar que mais difícil será reconstruir o sinal esparso. Portanto, como era intuitivo, o próximo teorema (CANDÈS; ROMBERG, 2007) mostra que quanto menor o valor da coerência mútua, melhor a propriedade de recuperação da matriz envolvida.

Teorema 4.2. *Dado $z \in \mathbb{R}^n$, suponha que a sequência de coeficientes x de z na base unitária $\Psi : n \times n$ é q-esparso. Selecione m medidas do domínio de Φ unitário uniformemente ao acaso. Então se*

$$m \geqslant C \cdot q \cdot n \cdot \mu(\Phi, \Psi)^2 \cdot \log(\frac{n}{p})$$

e

$$m \geqslant C' \cdot \log^2(\frac{n}{p})$$

para constantes positivas C, C', então com grande probabilidade excedendo $1-p$, o vetor x é a única solução para o problema de minimização na norma l_1 (4.8), com $A = R_m \Phi^T \Psi$, onde $R_m R_m^T = I_m$ e tendo A linhas ortonormais (4.15).

Para o desenvolvimento eficiente da implementação, faz-se uso da propriedade mais geral de uma matriz CS, propriedade essa da qual todas as matrizes CS satisfazem. Primeiro vamos enfraquecer a condição de ortonormalidade (4.15), assumindo que:

- As linhas da matriz A são aproximadamente ortonormais, ou seja,

$$\left\| AA^T - I_m \right\|_2 \leqslant \delta, \tag{4.17}$$

  para um δ pequeno.

Na Definição 4.1 de RIP, considera-se as colunas de A normalizadas (CANDÈS; ROMBERG; TAO, 2006), ou seja, as colunas que satisfazem RIP são quase ortonormais.

- Nessa tese vamos considerar que todo conjunto de q colunas de A são quase ortogonais, ou seja, $B^T B$ será próximo a $\frac{m}{n} I_q$, onde B é composto de q colunas arbitrárias de A. Formalmente, para toda matriz B temos:

$$\left\| \frac{n}{m} B^T B - I_q \right\|_2 \leqslant \delta_q, \tag{4.18}$$

  onde I_q é a matriz identidade de ordem q, com $q << m$.

Com essa nova reformulação nas condições da RIP (4.18), o Teorema 4.1 é reescrito como:

Teorema 4.3. *Todo vetor q-esparso $x \in \mathbb{R}^n$ satisfazendo $Ax = \hat{b}$ é a única solução de (4.8) se*

$$\delta_{2q} < \frac{3^{\frac{m}{n}}}{1 + 3^{\frac{m}{n}} + \sqrt{6}},$$

onde δ_{2q} é a constante mínima tal que (4.18) segue para todas $2q$ colunas da matriz A, denotado pela matriz B em (4.18).

A demonstração do Teorema 4.3 pode ser encontrada em (FOUNTOULAKIS, 2015).

Comparando os dois limitantes das constantes RIP dos Teoremas 4.1 e 4.3, observamos que o último limitante é menor (ver Figura 2.2 de (FOUNTOULAKIS, 2015)).

Contudo, a propriedade (4.18) e o Teorema 4.3 resultam em uma limitação da esparsidade máxima q, para a qual o problema (4.8) garante uma recuperação exata da solução esparsa de $Ax = \hat{b}$. Felizmente, os resultados dos Teoremas 4.1 e 4.3 são pessimistas. Em Blanchard, Cartis e Tanner (2011) mostra-se que as condições RIP da forma (4.14) e (4.18) fornecem os piores casos de δ_{2q} e consequentemente do nível de esparsidade q tal que o problema (4.8) garanta exata recuperação esparsa. É mostrado em Donoho e Tanner (2010) que o nível da esparsidade média máxima q, que é garantida ser reconstruída por (4.8), é muito maior que as mostradas nos Teoremas 4.1 e 4.3.

4.3 Dicionários

Vamos assumir que $\hat{x}$, solução de (4.6), possui uma imagem esparsa através de dicionários redundantes e coerentes $W \in E^{n\times l}$, onde $E = \mathbb{R}$ ou $E = \mathbb{C}$ e $n \leqslant l$. Dessa forma $W^*\hat{x}$ é esparso, onde $*$ denota o operador transposto conjugado. Se $W^*\hat{x}$ é esparso sob certas condições da matriz A e W, a solução ótima do problema:

$$\begin{aligned} &\text{minimizar} && \|W^*x\|_1 \\ &\text{sujeito a} && Ax = \hat{b}, \end{aligned} \tag{4.19}$$

é $\hat{x}$.

Se as medidas $\hat{b}$ estão contaminadas por ruído consideramos $b = \hat{b} + e$. Em aplicações reais $W^*\hat{x}$ pode não ser exatamente esparso, mas suas informações podem ser concentradas em apenas poucos componentes. Neste caso, sob certas condições das matrizes A e W, a solução ótima do problema:

$$\text{minimizar} \qquad \tau \|W^*x\|_1 + \frac{1}{2}\|Ax - b\|_2^2, \tag{4.20}$$

é provada ser uma boa aproximação para $\hat{x}$ para algum τ.

4.3.1 Propriedades das Matrizes A e W em CS

Como foi visto na seção anterior, a reconstrução do vetor x q-esparso depende da RIP. Vamos reformular a Definição 4.1.

Considere $W^*\hat{x}$ esparso igual a z, com W tendo linhas ortonormais, assim temos que:

1. $AWz = AWW^*\hat{x} = A\hat{x}$,
2. $Wz = WW^*\hat{x} = \hat{x}$.

A RIP adaptada a W é definida considerando a imagem de x esparsa através de W, ou seja, $z = W^*x$, com z esparso. Dessa forma, segue a nova definição para a RIP, chamada W-RIP.

Definição 4.2. *A Constante da Isometria Restrita (W-RIP) δ_q, sendo $\delta_q \geqslant 0$, de uma matriz $A \in \mathbb{R}^{m\times n}$ adaptada a $W \in E^{n\times l}$ é definida como o menor δ_q tal que*

$$(1-\delta_q)\left\|Wz\right\|_2^2 \leqslant \left\|AWz\right\|_2^2 \leqslant (1+\delta_q)\left\|Wz\right\|_2^2 \tag{4.21}$$

para todos os vetores z no máximo q-esparsos, $z \in \mathbb{E}^l$, onde $E = \mathbb{R}$ ou $\mathbb{C}$.

É provado em Candès et al. (2011) que se $W \in E^{n\times l}$ tem linhas ortonormais, com $n \leqslant l$, e se A e W satisfazem W-RIP com $\delta_{2q} \leqslant 8.0e-2$, então a solução x_τ obtida resolvendo (4.20) satisfaz:

$$\left\|x_\tau - \hat{x}\right\|_2 \leqslant C_0\left\|e\right\|_2 + C_1\frac{\left\|W^*x_\tau - (W^*\hat{x})_q\right\|_1}{\sqrt{q}}, \tag{4.22}$$

onde $(W^*\hat{x})_q$ é a melhor aproximação q-esparsa de $W^*\hat{x}$, C_0 e C_1 são constantes de módulo pequeno e que dependem apenas de δ_{2q}.

Isotropic total-variation (iTV), é um caso especial onde a matriz W não possui linhas ortonormais, assim, o resultado (4.22) não segue. Para iTV não existem condições em δ_{2q} tais que uma boa reconstrução é garantida. Contudo, existem resultados aos quais impõem diretamente restrições no número de medidas m. De uma forma sucinta, sabe-se que se $m \geqslant qlog(n)$ medidas lineares são adquiridas, para as quais as matrizes A e W satisfazem W-RIP para algum $\delta_q < 1$, então garantia de reconstrução similar aquelas em (4.22) são obtidas para iTV. Baseado no que foi mencionado anteriormente, a seguinte Hipótese é dada em Dassios, Fountoulakis e Gondzio (2015):

Hipótese 4.1. *O número de componentes diferentes de zero de W^*x_τ, denotando por q, e as dimensões l, m, n são tais que as matrizes A e W satisfaçam W-RIP para algum $\delta_{2q} < \frac{1}{2}$.*

Uma propriedade da matriz A é que suas linhas são aproximadamente ortonormais, ou seja,

$$\left\|AA^T - I_m\right\|_2 \leqslant \delta, \tag{4.23}$$

com $\delta \geqslant 0$ uma contante pequena, como já foi mencionado na seção anterior.

Por fim, vamos fazer uso da seguinte hipótese:

$$Ker(W^*) \cap Ker(A) = \{0\}\,.$$

Esta é uma hipótese realista e comumente usada, derivada da literatura de otimização (VAITER et al., 2013), a qual é necessária para que o problema (4.20) tenha uma única solução.

Vamos analisar essa hipótese. Suponha que existe um vetor $v \neq \overline{0}$, $v \in Ker(W^*)$ e $v \in Ker(A)$. Queremos:

$$\begin{array}{ll} \text{minimizar} & \|W^*x\|_1 \\ \text{sujeito a} & Ax = \hat{b} \end{array}\,. \tag{4.24}$$

Considere $\overline{x}$ uma solução do problema anterior, se $A\overline{x} = \hat{b}$, temos que:

$$A(\overline{x} + v) = A\overline{x} + Av = A\overline{x},$$

pois $v \in Ker(A)$. Além disso,

$$W^*(\overline{x} + v) = W^*\overline{x} + W^*v = W^*\overline{x},$$

pois $v \in KerW^*$. Dessa forma, não teríamos uma única solução.

Nesse capítulo discutimos a teoria de *Compressive Sensing* e apresentamos sua formulação, sendo uma por meio de dicionários e outra não. No capítulo seguinte apresentamos técnicas para suavizar funções não lineares, em especial a função $\|\cdot\|_1$ contida no problema de minimização (4.19).

Capítulo 5

Reformulação Primal-Dual por meio da Transformada Legendre-Fenchel

Neste capítulo a transformada Legendre-Fenchel (LF) é apresentada, mostrando que por meio dela podemos obter uma aproximação suave de uma função não suave. A ideia da aproximação suave de Moreau é mostrada, assim como exemplo comprovando que podemos obter aproximações com derivadas de todas as ordens ou apenas diferenciável de primeira ordem. Obtemos uma reformulação primal-dual do problema de interesse. Exemplos aplicados a determinadas funções também serão apresentados.

5.1 A Transformada Legendre-Fenchel

Transformações são um modo de mapear a função a um outro espaço, podendo resultar num melhor e mais fácil entendimento da função. Exemplos são a Transformada de Fourier e a Transformada de Laplace. A transformada Legendre-Fenchel é também um exemplo, a qual mapeia o espaço $(x, \psi(x))$ ao espaço de inclinações e ψ conjugadas, isto é $(y, \psi^*(y))$. Contudo, a transformada de Fourier consiste de uma integração com um kernel, já a transformada LF usa o supremo como o procedimento de transformação. Sob a hipótese que a transformação é reversível, uma forma é a dual da outra. Isto é expresso como:

$$(x, \psi(x)) \Leftrightarrow (y, \psi^*(x)),$$

sendo y a inclinação e $\psi^*(y)$ chamado o convexo conjugado da função $\psi(x)$. A função conjugada permite construir um problema dual que seja mais fácil de resolver que o primal. A conjugada Legendre-Fenchel é sempre convexa.

A transformada Legendre-Fenchel (ou Fenchel Conjugate (KAKADE; SHALEV-SHWARTZ; TEWARI, 2009)) é um modo de representar uma função convexa (ZIA; REDISH; MCKAY, 2009). A Transformada LF de uma função contínua, mas não necessariamente diferenciável, $\psi : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$, é definida como (HANDA et al., 2011):

$$\psi^*(y) = \sup_{x \in \mathbb{R}^n} \left\{ y^T x - \psi(x) \right\}. \tag{5.1}$$

O domínio de ψ^* é o conjunto de todas as inclinações das funções tangentes de ψ, ou seja, os sub-gradientes de ψ. A seguir é dada a definição de sub-gradiente (SIMÕES, 2013).

Definição 5.1. *Dada uma função convexa $f : \mathbb{R}^n \to \mathbb{R}$ e $x \in \mathbb{R}^n$, definimos o sub-diferencial $\partial f(x)$ de f em x como*

$$\partial f(x) := \{ v \in \mathbb{R}^n / f(x) + \langle v, y - x \rangle \leqslant f(y), \forall y \in \mathbb{R}^n \}.$$

Um elemento pertencente ao conjunto $\partial f(x)$ é dito ser um sub-gradiente de f em x.

A imagem de ψ^* é o conjunto de todos os interceptos negativos das funções tangentes a ψ. Geometricamente, isto significa que estamos interessados em encontrar um ponto x na função $\psi(x)$ tal que a inclinação y da reta que passa por $(x, \psi(x))$ tenha o intercepto máximo no eixo correspondente à imagem de ψ. Isto ocorre no ponto da curva tal que $\psi'(x) = y$.

Para um melhor entendimento, assuma por simplicidade que $\psi : \mathbb{R}^n \to \mathbb{R}$ é convexa. Temos que uma função f tangente a ψ no ponto x' é dada por $f(x) = y^T(x - x') + \psi(x')$, onde $y \in \partial\psi(x')$ é um sub-gradiente de ψ no ponto x'. Quando $x = 0$, a função tangente cruza o eixo perpendicular ao eixo x (e com a mesma origem), assim:

$$f(0) = -y^T x' + \psi(x') \Rightarrow -f(0) = y^T x' - \psi(x'),$$

então para cada x em (5.1), o valor de $y^T x - \psi(x)$ é igual a $-f(0)$, em que f é a função tangente a $\psi(x)$ no ponto x. O gráfico pode ser visto na Figura 2.

Assim para cada x, teremos um intercepto diferente, sendo que queremos o maior dentre todos.

Em Handa et al. (2011) é provado que a transformada LF é sempre convexa.

A seguir é dado um exemplo da transformada LF para a função norma l_2 (note que esta função é não diferenciável em zero).

$$f(y) = \|y\|_2,$$

$$f^*(z) = \sup_{y \in \mathbb{R}^n} \left\{ z^T y - \|y\|_2 \right\}.$$

Pela desigualdade de Cauchy-Schwarz, vemos que

$$\max_{\|z\|_2 \leqslant 1} z^T y \leqslant \max_{\|z\|_2 \leqslant 1} \|z\|_2 \|y\|_2 \leqslant \|y\|_2,$$

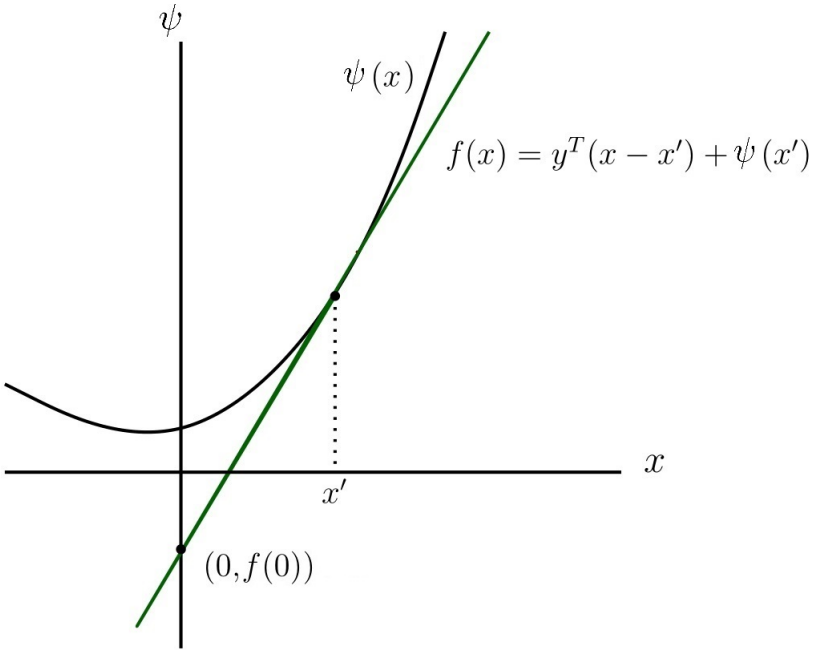

Figura 2 – Gráfico da função ψ e sua função tangente f no ponto x'.

tomando um z arbitrário, tal que $\|z\|_2 = 1$,

$$\max_{\|z\|_2 \leqslant 1} z^T y = \|y\|_2 .$$

Isso faz com que,

$$\max_{\|z\|_2 \leqslant 1} \left\{ z^T y - \|y\|_2 \right\} = 0 \quad \forall y \in \mathbb{R}^n.$$

Assim,

$$f^*(z) = \begin{cases} 0, & \text{se } \|z\|_2 \leqslant 1 \\ \infty, & \text{caso contrário} \end{cases} .$$

Exemplos de mais transformadas LF são dados em Handa et al. (2011). Em particular, gostaríamos de comentar a norma l_2 com

$$f(y) = \frac{1}{2} \|y\|_2 . \tag{5.2}$$

Escrevemos a transformada LF como:

$$f^*(z) = \sup_{y \in \mathbb{R}^n} \left\{ y^T z - \frac{1}{2} \|y\|_2^2 \right\} .$$

Podemos encontrar o ponto no qual a função atinge o valor máximo igualando o gradiente obtido por meio de derivadas direcionais a zero. Fazendo isso encontramos que o máximo se dá quando $y = z$, ou seja, quando $f^*(z) = \frac{1}{2} \|z\|_2^2$.

E lembrando que a norma Euclidiana é igual à norma dual dela (Apêndice A):

$$f^*(z) = \frac{1}{2}\|z\|_*^2 .$$

Dualidade é um modo de estudarmos uma mesma função de duas formas diferentes, chamadas primal e dual (HANDA et al., 2011). Muitas vezes é interessante um entendimento melhor da função que estamos interessados para uma melhor análise e estudo de seu comportamento, por exemplo, se ela é linear, se tem um bom comportamento em um dado domínio, entre outros.

5.2 Técnica de Suavização de *Moreau*

Estamos interessados em duas propriedades da transformada LF, que nos ajudarão na discussão da técnica de suavização:

- Para uma função convexa fechada ψ, a transformada LF ψ^{**} de ψ^* é igual a ψ :

$$\psi = \sup_y \left\{y^T x - \psi^*(y)\right\} = \psi^{**}. \tag{5.3}$$

- E,

$$\nabla\psi(x) = \arg\max_y \left\{y^T x - \psi^*(y)\right\}. \tag{5.4}$$

Note que a propriedade (5.4) implica que a função convexa ψ é diferenciável se, e somente se, o arg max retorna uma única solução. Se a transformada LF ψ^* da função ψ for fortemente convexa, a função ψ é diferenciável. Lembrando que:

Definição 5.2. *Uma função diferenciável f é chamada fortemente convexa com parâmetro $m > 0$, se a seguinte desigualdade segue para todos os pontos x, y em seu domínio (BERTSEKAS; NEDIC; OZDAGLAR, 2003):*

$$(\nabla f(x) - \nabla f(y))^T (x - y) \geqslant m\|x - y\|_2^2 ,$$

ou, de Nesterov (2004):

$$f(y) \geqslant f(x) + \nabla f(x)^T (y - x) + \frac{m}{2}\|y - x\|_2^2 ,$$

ou, para $t \in [0, 1]$:

$$f(tx + (1-t)y) \leqslant tf(x) + (1-t)f(y) - \frac{1}{2}mt(1-t)\|x - y\|_2^2 ,$$

ou, segundo Ivanov (1996), se a função

$$f(x) - \frac{m}{2}\|x\|_2^2$$

é convexa.

A técnica de suavização de Moreau força (5.4) a ter uma única solução, para isso é adicionada uma função fortemente convexa $\frac{\mu}{2}\|y\|_2^2 (\mu > 0)$ a ψ^*. Com isso, uma aproximação suave ψ_μ da função ψ é obtida por meio da propriedade (5.3):

$$\psi_\mu(x) = \sup_y \left\{ y^T x - \psi^*(y) - \frac{\mu}{2}\|y\|_2^2 \right\}. \tag{5.5}$$

O menor μ é a melhor aproximação da função ψ_μ a ψ (NESTEROV, 2005).

Podemos adicionar outras funções fortemente convexas, as quais resultam em diferentes aproximações suaves ψ_μ. A seguir é dado um exemplo.

Considere uma função $h(x) = \left\|\Omega^T x\right\|_2$, onde $\Omega \in \mathbb{R}^{n\times p}$. Vamos apresentar um exemplo da técnica de suavização de Moreau para esta função, como feito em Fountoulakis (2015).

A transformada LF da função $h(x) = \left\|\Omega^T x\right\|_2$ é:

$$h^*(y) = \sup_x \left\{ y^T x - \left\|\Omega^T x\right\|_2 \right\}. \tag{5.6}$$

Note que esse exemplo assemelha-se a (5.2), para o qual encontramos a transformada LF anteriormente.

Por meio das condições de otimalidade do problema (5.6), dadas por:

$$y = \Omega g, g \in \mathbb{R}^p \quad \|g\|_2 \leqslant 1, \tag{5.7}$$

obtemos,

$$h^*(\Omega g) = \begin{cases} 0, & \text{se } \|g\|_2 \leqslant 1 \\ +\infty, & \text{caso contrário} \end{cases}. \tag{5.8}$$

E de (5.3), (5.7) e (5.8) obtemos,

$$h(x) = \sup_y \left\{ y^T x - h^*(y) \right\} = \sup_{\|g\|_2 \leqslant 1} \left\{ g^T \Omega^T x \right\}. \tag{5.9}$$

Agora, note que o supremo em (5.9) não corresponde a uma solução única. Dessa forma, baseado em (5.4), a função h não é diferenciável.

Como temos interesse em tornar a função h suave, vamos redefinir h^* com uma função fortemente convexa, para isso, consideremos duas funções fortemente convexas:

$$\frac{\mu}{2}\|g\|_2^2, \tag{5.10}$$

e

$$\mu - \mu(1 - \|g\|_2^2)^{\frac{1}{2}}. \tag{5.11}$$

E agora:

1. Subtraindo (5.10) de (5.9), obtemos:

$$h_\mu(x) = \sup_{\|g\|_2 \leqslant 1} \left\{ g^T \Omega^T x - \frac{\mu}{2} \|g\|_2^2 \right\} = \begin{cases} \frac{\mu}{2} \left\|\Omega^T x\right\|_2^2, & \text{se} \quad \left\|\Omega^T x\right\|_2 \leqslant \mu \\ \left\|\Omega^T x\right\|_2 - \frac{\mu}{2}, & \text{se} \quad \left\|\Omega^T x\right\|_2 \geqslant \mu \end{cases}. \tag{5.12}$$

2. Subtraindo (5.11) de (5.9), obtemos:

$$h_\mu(x) = \sup_{\|g\|_2 \leqslant 1} \left\{ g^T \Omega^T x + \mu(1 - \|g\|_2^2)^{\frac{1}{2}} - \mu \right\} = \left(\mu^2 + \left\|\Omega^T x\right\|_2^2\right)^{\frac{1}{2}} - \mu. \tag{5.13}$$

Note que diferente de (5.12), que é apenas diferenciável em primeira ordem, (5.13) tem derivadas de qualquer ordem.

5.2.1 Funções Huber e Pseudo-Huber

No exemplo anterior estávamos trabalhando no conjunto dos números reais ($\mathbb{R}$), vamos estender agora para o conjunto dos números complexos ($\mathbb{C}$). Dada a função $\psi(x) = \|W^* x\|_1$, onde $W \in \mathbb{E}^{n \times l}$ e $\mathbb{E} = \mathbb{R}$ ou $\mathbb{C}$. Definimos $\Omega_i = [ReW_i, ImW_i] \in \mathbb{R}^{n \times 2}$, sendo que $Re(\cdot)$ retorna a parte real, e $Im(\cdot)$ a parte imaginária de um número complexo.

Podemos reescrever a função ψ da seguinte forma:

$$\psi(x) = \|W^* x\|_1 = \sum_{i=1}^{l} |W_i^* x| = \sum_{i=1}^{l} \left\|\Omega_i^T x\right\|_2. \tag{5.14}$$

Agora, vamos usar o exemplo da suavização da norma l_2 da Seção 5.2, com isso obtemos a seguinte aproximação:

$$\psi_\mu(x) = \sum_{i=1}^{l} h_\mu^i(x), \tag{5.15}$$

onde $h_\mu^i(x)$ pode ser (5.12) ou (5.13), usando a matriz Ω_i correspondente. Em particular, usando a aproximação (5.12) em (5.15), obtemos:

$$\psi_\mu(x) = \sum_{i=1}^{l} \begin{cases} \frac{\mu}{2} \left\|\Omega_i^T x\right\|_2^2, & \text{se} \quad \left\|\Omega_i^T x\right\|_2 \leqslant \mu \\ \left\|\Omega_i^T x\right\|_2 - \frac{\mu}{2}, & \text{se} \quad \left\|\Omega_i^T x\right\|_2 \geqslant \mu \end{cases}. \tag{5.16}$$

Usando a aproximação (5.13) em (5.15), obtemos:

$$\psi_\mu(x) = \sum_{i=1}^{l} \left((\mu^2 + \left\|\Omega_i^T x\right\|_2^2)^{\frac{1}{2}} - \mu \right). \tag{5.17}$$

Agora, da definição de Ω_i, temos que,

$$\psi_\mu(x) = \sum_{i=1}^{l} \begin{cases} \frac{\mu}{2}|W_i^* x|^2, & \text{se} \quad |W_i^* x| \leqslant \mu \\ |W_i^* x| - \frac{\mu}{2}, & \text{se} \quad |W_i^* x| \geqslant \mu \end{cases}, \tag{5.18}$$

é equivalente a (5.16). A equação (5.18) é conhecida como função Huber (FOUNTOULAKIS; GONDZIO, 2013). Temos também que (5.17) é equivalente a:

$$\psi_\mu(x) = \sum_{i=1}^{l}((\mu^2 + |W_i^* x|^2)^{\frac{1}{2}} - \mu), \tag{5.19}$$

a qual é conhecida como função pseudo-Huber.

Note então que, se quisermos utilizar tais propriedades dessas funções (Huber e pseudo-Huber) iremos, em nossas equações de interesse, substituir a função ψ pela ψ_μ mais conveniente.

Observamos que a função Huber é apenas diferenciável em primeira ordem, enquanto que a função pseudo-Huber é suave. Uma comparação das três funções é apresentada na Figura 3, retirada de Fountoulakis e Gondzio (2013).

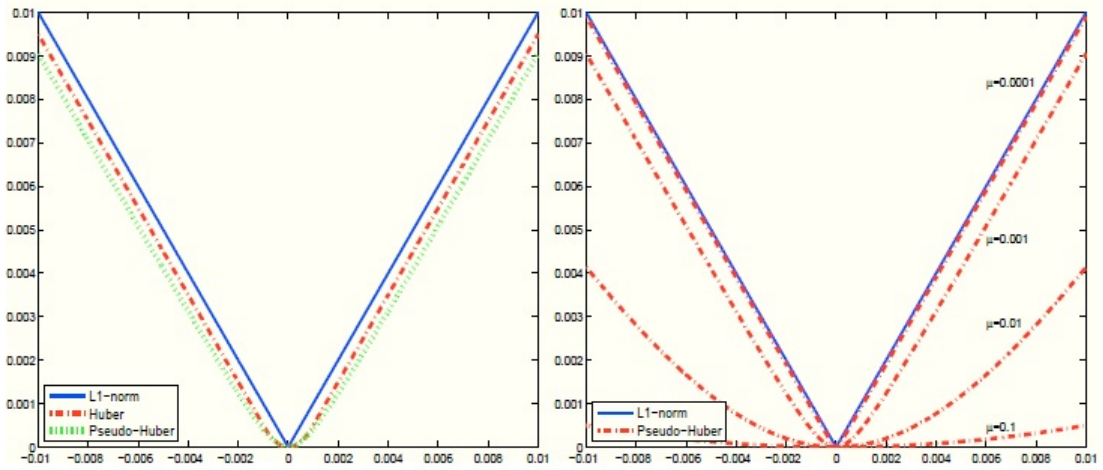

Figura 3 – Nota-se que a função pseudo-Huber converge para a norma l_1 quando μ tende a zero. Figura retirada de Dassios, Fountoulakis e Gondzio (2015).

5.3 Reformulação Primal-Dual

Com o objetivo de trabalhar com uma função próxima à desejada, mas que possua derivadas de todas as ordens, vamos reescrever o problema original em sua forma primal-dual utilizando a função pseudo-Huber ψ_μ no lugar de ψ. As vantagens de tal aproximação são listadas a seguir (FOUNTOULAKIS; GONDZIO, 2013):

- Disponibilidade de informações de segunda ordem, devido à diferenciabilidade da função pseudo-Huber.
- Liberdade ao uso de métodos iterativos para o cálculo das direções de descida, as quais levam em conta a curvatura do problema, tal como o Método dos Gradientes Conjugados.

5.3.1 Reformulação Primal-Dual por meio da Tranformada LF

Estamos interessados no problema:

$$\min f_\tau(x) := \tau\psi(x) + \varphi(x), \tag{5.20}$$

onde $x \in \mathbb{R}^n$ e τ é um parâmetro positivo. Sendo $\psi : \mathbb{R}^n \to \mathbb{R}$ uma função convexa possivelmente não suave e $\varphi : \mathbb{R}^n \to \mathbb{R}$ uma função convexa, podemos aplicar a transformada LF para obter uma reformulação primal-dual do problema (5.20). Em particular, a função não suave ψ em (5.20) pode ser substituída por sua forma equivalente (5.3). Dessa forma, o problema (5.20) é reescrito como:

$$\min_x(\sup_y \left\{\tau y^T x - \tau\psi^*(y)\right\} + \varphi(x)), \tag{5.21}$$

onde y são as variáveis duais. Note que (5.3) é uma reformulação dual de ψ, dessa forma (5.21) é uma reformulação primal-dual de (5.20).

5.3.2 Reformulação Primal-Dual do Problema aproximado pela Função Pseudo-Huber

Gostaríamos de resolver o problema (5.20), mas com o objetivo de obter uma função que possua todas as derivadas, vamos regularizar o problema (5.20) substituindo a função ψ pela função pseudo-Huber ψ_μ, obtendo:

$$\min \tau\psi_\mu(x) + \varphi(x), \tag{5.22}$$

onde $\psi_\mu(x)$ é a função pseudo-Huber definida em (5.17). Usando (5.13) vamos obter a seguinte formulação primal-dual equivalente a (5.17):

$$\psi_\mu(x) = \sum_{i=1}^{l}(\sup_{g_i\in\mathbb{R}^2, \|g_i\|_2\leqslant 1} \left\{g_i^T\Omega_i^T x + \mu(1 - \|g_i\|_2^2)^{\frac{1}{2}} - \mu\right\}).$$

Sabemos que todo vetor bidimensional g_i pode ser reescrito como um número complexo, e usando a definição $\Omega_i = [ReW_i, ImW_i] \in \mathbb{R}^{n\times 2}$, obtemos:

$$\psi_\mu(x) = \sup_{g\in\mathbb{C}^l, \|g\|_\infty\leqslant 1} \left\{ Re(\overline{g}^* W^*)x + \sum_{i=1}^{l}(\mu(1 - |g_i|^2)^{\frac{1}{2}} - \mu) \right\},$$

onde g são as variáveis duais e a barra denota o conjugado do número complexo. Dessa forma, a formulação primal-dual do problema (5.22) será:

$$\min_{x}((\sup_{g\in\mathbb{C}^l,\|g\|_\infty\leqslant 1}\left\{\tau Re(\overline{g}^* W^*)x + \tau\sum_{i=1}^{l}(\mu(1-|g_i|^2)^{\frac{1}{2}}-\mu)\right\} + \varphi(x)). \tag{5.23}$$

Nessa capítulo apresentamos técnicas de suavização. No próximo capítulo retomaremos ao problema (4.19) em que a formulação primal-dual dada por (5.23) será incorporada ao problema. Um método conhecido da literatura para a sua resolução é discutido.

Capítulo 6

Primal-Dual Newton Conjugate Gradients (pdNCG) para *Compressive Sensing*

Nesse capítulo vamos apresentar o Método *Primal-Dual Newton Conjugate Gradients* (pdNCG) de Fountoulakis (2015) especializado para problemas *Compressive Sensing*. O método tem o objetivo de encontrar a solução do seguinte problema para W^*x esparso:

$$\min_x \tau \left\|W^*x\right\|_1 + \frac{1}{2}\left\|Ax-b\right\|_2^2, \tag{6.1}$$

onde $W \in E^{n\times l}$, $E = \mathbb{R}$ ou $\mathbb{C}, n \leqslant l, A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^m, \tau > 0$, sendo que o asterisco denota o operador transposto conjugado.

Uma técnica de pré-condicionamento com baixo custo computacional e comprovadamente efetiva é apresentada. Em toda iteração do pdNCG, a maior parte do custo computacional está na solução inexata de um sistema linear com uma matriz mal condicionada. É provado que o pré-condicionador agrupa os autovalores da matriz do sistema linear em torno de 1 (DASSIOS; FOUNTOULAKIS; GONDZIO, 2015). Dessa forma, os sistemas lineares podem ser resolvidos eficientemente usando um método iterativo.

Nesse trabalho, a implementação do algoritmo proposto em Dassios, Fountoulakis e Gondzio (2015) é aplicada.

6.1 Reformulação por meio da Função Pseudo-Huber

Como mencionado anteriormente, gostaríamos de trabalhar com funções cujas informações sejam mais fáceis de serem analisadas. Assim, a não diferenciabilidade da norma l_1 é contornada reformulando o problema (6.1) pelo seguinte problema aproximado:

$$min f_\tau^\mu(x) := \tau\psi_\mu(x) + \frac{1}{2}\|Ax-b\|_2^2, \tag{6.2}$$

onde

$$\psi_\mu(x) = \sum_{i=1}^{l}((\mu^2 + |W_i^* x|^2)^{\frac{1}{2}} - \mu).$$

6.2 Derivadas Parciais de Primeira e Segunda Ordem

Aqui são apresentados os vetores gradientes e as matrizes Hessianas das funções de interesse, como feito em Fountoulakis (2015).

O gradiente da função pseudo-Huber em (5.19) é:

$$\nabla\psi_\mu(x) = Re(WDW^*)x, \tag{6.3}$$

onde $D := diag(D_1, D_2, ..., D_l)$, com

$$D_i := (\mu^2 + |y_i|^2)^{-\frac{1}{2}} \quad \forall i = 1, 2, 3, ...l, \tag{6.4}$$

e $y = [y_1, y_2, ..., y_l]^T := W^*x$. A matriz Hessiana de ψ_μ é:

$$\nabla^2\psi_\mu(x) := \frac{1}{4}(W\hat{Y}W^* + \overline{W}\hat{Y}\overline{W}^* + W\tilde{Y}\overline{W}^* + \overline{W\hat{Y}}W^*), \tag{6.5}$$

onde $\hat{Y} := diag\left[\hat{Y}_1, \hat{Y}_2, ..., \hat{Y}_l\right]$, $\tilde{Y} := diag\left[\tilde{Y}_1, \tilde{Y}_2, ..., \tilde{Y}_l\right]$ e

$$\hat{Y}_i := \mu^2 D_i^3 + D_i, \qquad \tilde{Y}_i := -y_i^2 D_i^3, \quad i = 1, 2, ..., l. \tag{6.6}$$

O gradiente da função f_τ^μ em (6.2) é:

$$\nabla f_\tau^\mu(x) = \tau\nabla\psi_\mu(x) + A^T(Ax-b).$$

Sendo que, a matriz Hessiana de f_τ^μ é dada por:

$$\nabla^2 f_\tau^\mu(x) = \tau\nabla^2\psi_\mu(x) + A^TA. \tag{6.7}$$

6.3 Formulação Primal-Dual

Reformulando o problema (6.2) por meio de (5.23), com $\psi(x) = \frac{1}{2}\|Ax-b\|_2^2$, obtemos a formulação primal-dual seguinte:

$$min \sup_{g\in\mathbb{C}^l, \|g\|_\infty \leqslant 1} \left\{\tau Re(\overline{g}^* W^*)x + \tau\sum_{i=1}^{l}(\mu(1-|g_i|^2)^{\frac{1}{2}} - \mu)\right\} + \frac{1}{2}\|Ax-b\|_2^2. \tag{6.8}$$

As condições de otimalidade de (6.2) são:

$$\tau\nabla\psi_\mu(x) + A^T(Ax-b) = \tau Re(WDW^*)x + A^T(Ax-b) = 0. \tag{6.9}$$

Já as condições de otimalidade de (6.8) são:

$$\begin{aligned} \tau Re(W\overline{g}) + A^T(Ax-b) &= 0 \\ D^{-1}\overline{g} &= W^*x \end{aligned} \quad . \tag{6.10}$$

Sendo D definido em (6.4) e $g \in E^l$.

Note que de (6.8), $(1-|g_i|^2) \geqslant 0$, pois caso contrário, obteríamos uma solução no conjunto dos números complexos, e dessa forma:

$$1-|g_i|^2 \geqslant 0 \quad \forall i \Rightarrow |g_i|^2 \leqslant 1 \quad \forall i \Rightarrow \|g\|_\infty \leqslant 1.$$

Ou mais, da definição da matriz D em (6.4) e da segunda equação de (6.10), temos:

$$D^{-1}\overline{g} = W^*x \Rightarrow \overline{g} = DW^*x \Rightarrow |\overline{g}_i| = |D_i(W^*x)_i| =$$
$$= |(\mu^2 + |(W^*x)_i|^2)^{-\frac{1}{2}}(W^*x)_i| = \frac{1}{|(\mu^2+|(W^*x)_i|^2)^{\frac{1}{2}}|}|(W^*x)_i| \leqslant \frac{|(W^*x)_i|}{|(W^*x)_i|} = 1 \quad \forall i$$
$$\Rightarrow \|g\|_\infty \leqslant 1.$$

Dessa forma, a restrição $\|g\|_\infty \leqslant 1$ é redundante para qualquer x e g satisfazendo (6.10).

Note que as condições de otimalidade (6.10) foram obtidas de (6.9), isso é feito comparando a primeira equação de (6.10), obtida por meio de $\nabla_x \psi_\mu$, com (6.9); fazendo isso obtemos:

$$(WDW^*)x = W\overline{g} \Rightarrow W(DW^*x) = W\overline{g}.$$

Portanto, definindo $\overline{g} = DW^*x$ ou $D^{-1}\overline{g} = W^*x$, juntamente com $\nabla_x \psi_\mu$ (primeira equação de (6.10)), obtemos (6.9).

As condições de primeira ordem do problema primal (6.2) foram apresentadas em (6.9). Dessa forma, poderíamos simplesmente aplicar o Método *Newton Conjugate Gradients* a fim de solucionar o problema de interesse. Mas tem sido notado (CHAN; CHAN; ZHOU, 1995; CHAN; GOLUB; MULET, 1999) que a linearização de $\nabla\psi_\mu$ para o Método de *Newton Conjugate Gradients* pode não ser uma boa aproximação de $\nabla\psi_\mu$ próximo a uma solução ótima. Para sanar este problema os autores Chan, Golub e Mulet (1999) sugerem resolver as condições de otimalidade de (6.10) do problema primal-dual (6.8). O motivo é apresentado a seguir.

Considere $[\cdot]_{ij}$ o operador que retorna o elemento da linha i e coluna j da matriz, e $[\cdot]_i$ o operador que retorna o elemento do vetor que encontra-se na posição i. A linearização da segunda equação de (6.10), isto é, $\overline{g}_i / [D]_{ii} - [W^*x]_i = 0, \quad \forall i = 1,2,3,...,m$, é de muito melhor qualidade que a linearização de $[\nabla\psi_\mu(x)]_i$ para $\mu \approx 0$ e $[W^*x]_i \approx 0$ (uma situação inevitável, já que para μ pequeno uma solução ótima de

(6.1) é esperada ser aproximadamente esparsa). Observe que para μ de módulo pequeno e $[W^*x]_i \approx 0$, o gradiente $[\nabla\psi_\mu(x)]_i$ está próximo a ser indefinido, e sua linearização é esperada ser inexata. Ora, como $\nabla\psi_\mu(x) = Re(WDW^*)x$, com $D := diag(D_1, D_2, ..., D_l)$, onde $D_i := (\mu^2 + |(W^*x)_i|^2)^{-\frac{1}{2}}$, considere o caso $\mu = 0$ e $W^* = I$, como para $(W^*x)_i = 0$ a função não é definida, teríamos uma função descontínua, com

$$[\nabla\psi_\mu(x)]_i = \begin{cases} 1, & \text{para} \quad x_i > 0 \\ -1, & \text{para} \quad x_i < 0 \end{cases}. \tag{6.11}$$

Agora note que, $g_i/[D]_{ii} - x_i$ como função de x_i, está bem definida para $\mu \approx 0$ e $x_i \approx 0$.

Como mencionado em Fountoulakis (2015), justificativas empíricas para a vantagem em trabalhar com o problema primal-dual, são dadas na Seção 3 de Chan, Golub e Mulet (1999), assim como os resultados obtidos em Fountoulakis (2015).

Em suma, resolvemos o problema utilizando as condições de otimalidade primal-dual (6.10).

6.4 Construção do Método

Com a finalidade de trabalhar no campo dos números reais, podendo obter condições de otimalidade na análise real, as condições de otimalidade (6.10) são convertidas para o caso real. A conversão é feita considerando $W = ReW + \sqrt{-1}ImW$ e $g = g_{re} + \sqrt{-1}g_{im}$, sendo que a função $Im(\cdot)$ retorna a parte imaginária de um número. As condições de otimalidade com variáveis reais são:

$$\begin{aligned} &\tau Re(Wg_{re} + Img_{im}) + A^T(Ax - b) = 0 \\ &D^{-1}g_{re} = ReW^Tx \\ &D^{-1}g_{im} = ImW^Tx \end{aligned}. \tag{6.12}$$

Nas iterações de pdNCG, as direções são calculadas resolvendo a seguinte linearização das restrições de igualdade em (6.12):

$$\begin{aligned} &B\Delta x = -\nabla f_\tau^\mu(x) \\ &\Delta g_{re} = D(I - B_1)ReW^T\Delta x + DB_2ImW^T\Delta x - g_{re} + DReW^Tx \\ &\Delta g_{im} = D(I - B_4)ImW^T\Delta x + DB_3ReW^T\Delta x - g_{im} + DImW^Tx \end{aligned}, \tag{6.13}$$

onde

$$B := \tau\widetilde{B} + A^TA, \quad \text{e} \tag{6.14}$$

$\widetilde{B} := ReWD(I-B_1)ReW^T + ImWD(I-B_4)ImW^T + ReWDB_2ImW^T + ImWB_3DReW^T$,

e B_i, $i = 1, 2, 3, 4$ são matrizes diagonais com componentes iguais a:

$$\begin{aligned} &[B_1]_{ii} := D_i[g_{re}]_iReW_i^Tx, \qquad &&[B_2]_{ii} := D_i[g_{re}]_iIm_i^Tx, \\ &[B_3]_{ii} := D_i[g_{im}]_iReW_i^Tx, \qquad &&[B_4]_{ii} := D_i[g_{im}]_iImW_i^Tx \end{aligned}.$$

Fountoulakis (2015) observa que apesar de em certas condições a matriz B em (6.14) ser definida positiva, no caso de W ser complexa, a matriz B não é simétrica. Os autores propõem evitar o problema da matriz B não ser simétrica, como é sugerido em Chan, Golub e Mulet (1999), e aplicam CG para resolver (6.13).

Assim, o sistema (6.13) é reescrito como:

$$\begin{aligned}
&\hat{B}\Delta x = -\nabla f_\tau^\mu(x) \\
&\Delta g_{re} = D(I - B_1)ReW^T\Delta x + DB_2ImW^T\Delta x - g_{re} + DReW^T x \\
&\Delta g_{im} = D(I - B_4)ImW^T\Delta x + DB_3ReW^T\Delta x - g_{im} + DImW^T x
\end{aligned} \quad . \tag{6.15}$$

onde

$$\hat{B} := \tau \text{sym}(\tilde{B}) + A^T A, \tag{6.16}$$

e $\text{sym}(\tilde{B}) := \frac{1}{2}(\tilde{B} + \tilde{B}^T)$. A matriz $\text{sym}(\tilde{B})$ é a matriz simétrica mais próxima de $\tilde{B}$ de acordo com a norma de Frobenius.

O método PCG termina quando:

$$\left\| \hat{B}\Delta x + \nabla f_\tau^\mu(x) \right\| \leqslant \eta \left\| \nabla f_\tau^\mu(x) \right\|_2$$

é satisfeito para $\eta \in [0, 1)$. A restrição $\|g\|_\infty \leqslant 1$ é incluída projetando-se $g_{re} + \Delta g_{re} + \sqrt{-1}(g_{im} + \Delta g_{im})$ no espaço $\{x : \|x\|_\infty \leqslant 1\}$.

O pseudo-código de pdNCG é apresentado em Fountoulakis (2015).

Como a eficiência computacional de pdNCG depende do número de condição da matriz $\hat{B}$ em (6.16), um pré-condicionador para $\hat{B}$ é desenvolvido.

6.5 Pré-condicionamento

Na Observação A.2 do artigo de Dassios, Fountoulakis e Gondzio (2015) é mencionado que a distância w entre duas soluções $x_c := arg\ min f_\tau(x)$ e $x_{c,\mu} := arg\ min f_\tau^\mu(x)$ pode ser arbitrariamente pequena para valores suficientemente pequenos de μ. Sabemos também, que de acordo com a Hipótese 4.1, $W^* x_\tau$ é q-esparso. Dessa forma, a Observação A.2 implica que $W^* x_{\tau,\mu}$, é aproximadamente q-esparso com componentes quase nulas de $\mathcal{O}(w)$. Uma consequência do que é apresentado, é que os componentes de $W^* x_{\tau,\mu}$ são divididos em dois conjuntos disjuntos:

$$\mathcal{B} := \{i \in \{1, 2, ..., l\} / \ |W_i^* x_{\tau,\mu}| >> \mathcal{O}(w)\}, \quad |\mathcal{B}| = q = |sup(W^* x_\tau)|,$$

$$\mathcal{B}^c := \{i \in \{1, 2, ..., l\} / \ |W_i^* x_{\tau,\mu}| \approx \mathcal{O}(w)\}, \quad |\mathcal{B}^c| = l - q.$$

Note que a matriz $\nabla^2\psi_\mu(W^*x_{\tau,\mu})$, em (6.5), é influenciada pelo comportamento de $W^*x_{\tau,\mu}$. Os componentes diagonais da matriz D, definida em (6.4), e que é usada no calculo de $\nabla^2\psi_\mu(W^*x_{\tau,\mu})$, são divididos em dois conjuntos disjuntos. Os conjuntos são divididos da seguinte forma, um é formado por q elementos diferentes de zero, muito menores que $\mathcal{O}(\frac{1}{w})$, e o outro (que é formado pela maioria dos componentes), é composto por $l-q$ elementos de $\mathcal{O}\left(\frac{1}{w}\right)$,

$$D_i << \mathcal{O}\left(\frac{1}{w}\right) \quad \forall i \in \mathcal{B}, \quad e \quad D_i = \mathcal{O}\left(\frac{1}{w}\right) \quad \forall i \in \mathcal{B}^c. \tag{6.17}$$

Assim, quando os pontos estão próximos da solução, ou seja, perto de $x_{\tau,\mu}$, e μ é suficientemente pequeno, a matriz $\nabla^2 f_\tau^\mu(x)$, em (6.7), consiste da matriz dominante $\tau\nabla^2\psi_\mu(x)$ e, da matriz A^TA, que por (4.23), possui o valor do maior autovalor não expressivo. Como,

$$\lambda_{max}(A^TA) = \lambda_{max}(AA^T),$$

se em (4.23), λ não é uma constante de valor grande, então $\lambda_{max}(A^TA) \leqslant 1+\delta$. Como a matriz simétrica $sym(\widetilde{B})$, em (6.16) tende a matriz $\nabla^2\psi_\mu(x)$, a medida que $x \to x_{\tau,\mu}$, a matriz $sym(\widetilde{B})$ é a matriz dominante em $\widehat{B}$. Pelo que foi apresentado, Fountoulakis (2015) propõe a seguinte técnica de pré-condicionamento: a matriz A^TA em (6.7) é substituída pela matriz identidade escalada, ρI_n, em que ρ é maior que zero, e a matriz dominante $sym(\widetilde{B})$ é mantida. Fundamentado no que foi observado, o seguinte pré-condicionador é proposto:

$$\widetilde{N} := \tau sym(\widetilde{B}) + \rho I_n. \tag{6.18}$$

Para maiores detalhes do pré-condicionador, assim como resultados teóricos úteis para a análise do comportamento do espectro da matriz pré-condicionada $\widehat{B}$, consultar Fountoulakis (2015).

6.6 Método da Continuação

Como foi visto na seção anterior, um pré-condicionador adequado foi desenvolvido, para melhorar as propriedades espectrais do sistema. Nos estágios iniciais do método, o condicionamento da matriz $\widehat{B}$ é controlado sem o uso do pré-condicionador, que só passa a ser utilizado quando o valor do parâmetro de suavização μ é menor ou igual a 10^{-4}, ou seja, o pré-condicionador é utilizado apenas nos estágios finais. Esse controle é realizado por meio do Método da Continuação (NOCEDAL; WRIGHT, 2006; ALLGOWER; GEORG, 2003). Para uma breve apresentação do método, suponha que queremos resolver um problema da forma $F(x) = 0$, sendo x^* a solução desconhecida, e $T(x,\beta)$ homotopia de $F(x)$, em que $T : \mathbb{R}^n \times [0,1] \to \mathbb{R}^n$ e

$$T(x,\beta) = \beta F(x) + (1-\beta)G(x),$$

em que $G(x)$ tem solução conhecida. O Método da Continuação auxilia a determinar uma forma de passar da solução conhecida $x(0)$ de $T(x,0) = 0$, à solução desconhecida $x(1) \approx x^*$ de $T(x,1) = 0$, a qual resolve $F(x) = 0$.

Dessa forma, por meio do Método da Continuação resolvemos uma sequência de subproblemas que são mais facilmente tratáveis ao invés de resolvermos diretamente o problema (6.2). Na implementação em questão, cada subproblema é parametrizado por τ e μ simultaneamente. Dados $\widetilde{\tau}$ e $\widetilde{\mu}$ sendo os parâmetros finais para o qual o problema (6.2) deve ser resolvido, o número de iterações do método é dado por

$$\upsilon := \left\lfloor max(\mid log_{10}(\frac{1}{\widetilde{\tau}}) \mid, \mid log_{10}(\frac{1}{\widetilde{\mu}}) \mid) \right\rfloor .$$

Definem-se os parâmetros iniciais τ^0 e μ^0 iguais a $1.0e-1$ sempre que $\upsilon \geqslant 2$ e os intervalos $\left[\tau^0, \widetilde{\tau}\right]$ e $\left[\mu^0, \widetilde{\mu}\right]$ são divididos em υ subintervalos iguais em escala logarítmica.

Sugerimos ao leitor a Seção 5.7 de Fountoulakis (2015) para mais detalhes.

Capítulo 7

Método Proposto e Experimentos Computacionais

No Capítulo 6 foi apresentado o método *Primal-Dual Newton Conjugate Gradients* (pdNCG), sendo que seu desenvolvimento completo pode ser encontrado em Dassios, Fountoulakis e Gondzio (2015). Na nossa primeira abordagem, o pré-condicionador proposto para os problemas *Compressive Sensing* que baseia-se no pré-condicionador fator separador, ao qual denominamos pseudo fator separador, discutido na primeira parte da tese é apresentado na Seção 7.1. Em seguida expomos os experimentos numéricos realizados para essa abordagem. Com a finalidade de obter melhores resultados computacionais, nossa segunda abordagem consiste em duas implementações, a primeira implementação considerando uma modificação no cálculo das direções primal-dual é exposta na Subseção 7.9.1. Desenvolvemos também um segundo método, em que novas condições de otimalidade são consideradas, e que pode ser encontrado na Subseção 7.9.2.

7.1 Primeira Abordagem

O método que apresentamos nesse capítulo, que denotaremos por pdNCGs, baseia-se na aplicação da fatoração incompleta de Cholesky, na matriz do pré-condicionador sugerido por Fountoulakis e que vimos no capítulo anterior. O método tem como base a ideia de que, assim como o pré-condicionador Separador apresentado no Capítulo 2, o pré-condicionador apresentado em Dassios, Fountoulakis e Gondzio (2015), também tem um melhor desempenho nas iterações finais do método, ou seja, próximo à solução. Assim, dado o pré-condicionador (6.18), é aplicada a fatoração incompleta de Cholesky na matriz $\tilde{N}$, e o fator obtido é o pré-condicionador que iremos utilizar. Dessa forma, sendo $\hat{L}$ o fator obtido pela fatoração incompleta de Cholesky da matriz $\tilde{N}$, a matriz pré-condicionada tem a forma $\hat{L}^{-1}\tilde{N}\hat{L}^{-T}$.

Para realizar a fatoração incompleta de Cholesky, usamos a função ichol, do

Matlab. Nela utilizamos a retirada por tolerância (*drop tolerance*), com valor igual a 10^{-3}. Também é usado *shift* na diagonal, com um valor α apropriado.

Nos testes realizados na próxima seção, além da mudança de pré-condicionador, algumas alterações no código dos testes foram realizadas com o objetivo de tornar pdNCGs mais eficiente.

7.2 Experimentos Numéricos

Os experimentos numéricos foram implementados em Matlab R2014a, em um sistema operacional Microsoft Windows 10 com Intel® Core(TM) i7-5500U 2.40GHz e 8 GB de memória RAM. O código pdNCG implementado por Dassios, Fountoulakis e Gondzio (2015) é utilizado.

As imagens utilizadas para os testes são: House e Peppers (Figura 6), Lena e Fingerprint (Figura 7), Boat e Barbara (Figura 8), e Shepp-Logan (Figura 9), que são imagens padrão da área de processamento de imagens; e também Dice, Ball e Cup (Figura 4), Letter e Logo (Figura 5), amostradas usando uma câmera *single-pixel* (DUARTE et al., 2008). No total, elas somam 12 imagens, sendo House e Peppers de 256×256 pixels, Lena, Fingerprint, Boat e Barbara de 512×512 pixels. O tamanho da imagem Shepp-Logan varia conforme o experimento. As imagens Dice, Ball, Cup, Letter e Logo são de 64×64 pixels. Todos os problemas de reconstrução de imagem nessa tese são modelados usando *Isotropic total-variation* (iTV), a qual explora o fato que imagens digitais frequentemente tem pixels que variam lentamente, com exceção ao longo das bordas.

Seja p_v o número de pixels verticais da imagem a ser reconstruída, e p_h o número de pixels horizontais. Assuma, por simplicidade que a imagem é quadrada, assim $p = p_v = p_h$; e que a imagem é utilizada em forma de vetor, isto é, ao invés de uma imagem de $p \times p$, nós temos uma imagem em forma de vetor de tamanho $p^2 \times 1$, em que as colunas da imagem são ordenadas uma depois da outra. Em nosso trabalho, para iTV, a matriz $W \in \mathbb{C}^{n\times n}$ no problema (6.1) é quadrada com $n = p^2$, complexa e linearmente dependente (LD), pois $rank(W) = n - 1$.

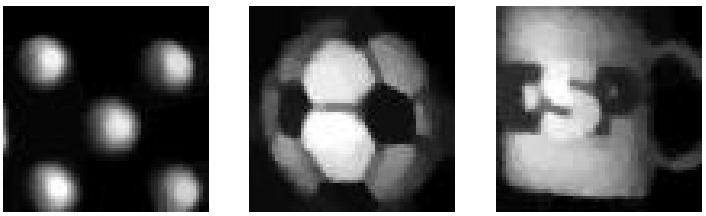

Figura 4 – Dice, Ball e Cup, extraídos de http://dsp.rice.edu/cscamera.

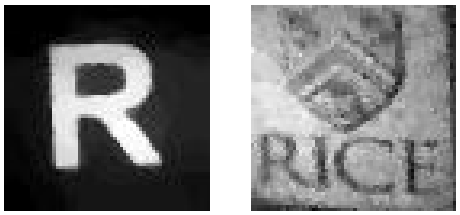

Figura 5 – Letter e Logo, extraídos de http://dsp.rice.edu/cscamera.

Figura 6 – House e Peppers, com 256^2 pixels.

Figura 7 – Lena e Fingerprint, com 512^2 pixels.

Figura 8 – Boat e Barbara, com 512^2 pixels.

Figura 9 – Shepp-Logan, com o tamanho definido no experimento.

Buscamos demonstrar a eficiência do método proposto, pdNCGs, comparando com quatro métodos de primeira ordem de estado da arte: TFOCS - Templates for First-Order Conic Solvers (BECKER; CANDÈS; GRANT, 2011), que resolve dois tipos de problemas, um sem restrição (TFOCS_unc) e outro com restrição TFOCS_con, TVAL3 - Total Variation minimization by Augmented Lagrangian and Alternating direction Algorithms (LI et al., 2013), TwIST - Two-step Iterative Soft Thresholding (BIOUCAS-DIAS; FIGUEIREDO, 2007), e o método pdNCG, que é o método que modificamos em busca de maior eficiência. Os métodos pdNCG, TFOCS_unc, TVAL3 e TwIST resolvem o problema 6.1, enquanto TFOCS_con resolve:

$$\begin{array}{ll} \text{minimizar}_x & \|W^* x\|_1 + \dfrac{\mu_{T_2}}{2}\left\|x - x^0\right\|_2^2, \\ \text{sujeito a} & \|Ax - b\|_2 \leqslant \varepsilon \end{array} \tag{7.1}$$

em que μ_{T_2} é uma constante positiva e ε é maior que zero. Assim, ajustes foram realizados nos parâmetros das formulações dos problemas, com a finalidade de fazer com que os métodos resolvam problemas similares. Para maiores detalhes a respeito dos parâmetros correspondentes a cada método, consultar (DASSIOS; FOUNTOULAKIS; GONDZIO, 2015). A implementação dos métodos de primeira ordem de estado da arte foram feitas por Dassios, Fountoulakis e Gondzio (2015).

Medimos a qualidade da imagem reconstruída usando a função *peak-signal-to-noise-ratio* (PSNR), dada por

$$PSNR = 10\ log_{10}\left(\frac{peakval^2}{MSE}\right),$$

em que *peakval* é o valor máximo possível do pixel, por exemplo, no caso de uma imagem de oito bits, o valor de *peakval* é 255, como no nosso caso as imagens são em preto e branco, *peakval* tem valor 1. MSE (erro quadrático médio) é a média dos quadrados dos erros, entre a solução da imagem original e a corrompida.

Os métodos pdNCGs, pdNCG, TVAL3 e TwIST terminam quando o PSNR da solução são iguais ou maiores que o PSNR da solução obtida por TFOCS_unc, isso garante que os métodos terminem quando uma solução de mesma, ou igual qualidade da TFOCS_unc, seja obtida.

7.3 Desempenho em relação ao número de medidas

O primeiro teste foi realizado com a imagem Ball. Nesse experimento, a matriz $A \in \mathbb{R}^{m\times n}$, é uma base parcial de Walsh (GOLUBOV; EFIMOV; SKVORTSOV, 2012), que toma valores 0 ou 1. Nele analisamos como a imagem é recuperada, por meio dos métodos pdNCG e pdNCGs, conforme diminuímos o número de medidas m. A porcentagem de medidas é mostrada sob as Figuras 10 e 11. Por exemplo, 40% significa que $m \approx \dfrac{2n}{5}$,

em que n é o número de pixels da imagem a ser reconstruída. A Figura 10 apresenta a reconstrução obtida para o método pdNCG, e a Figura 11 para nosso método, pdNCGs.

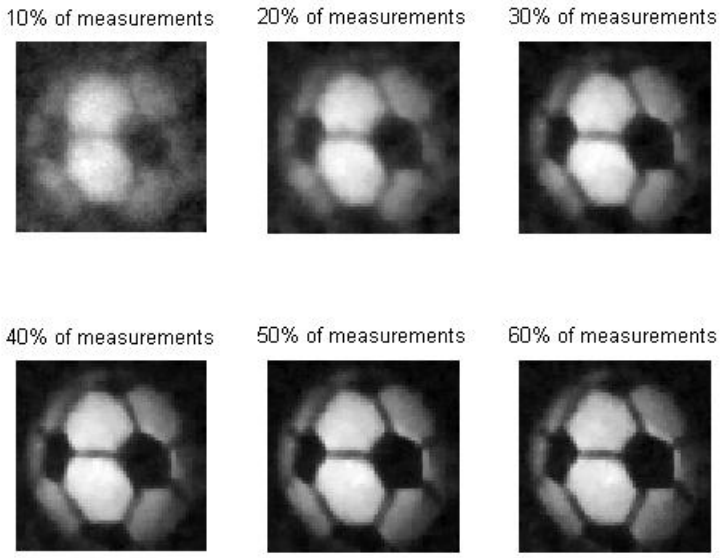

Figura 10 – Ball obtida por meio de pdNCG.

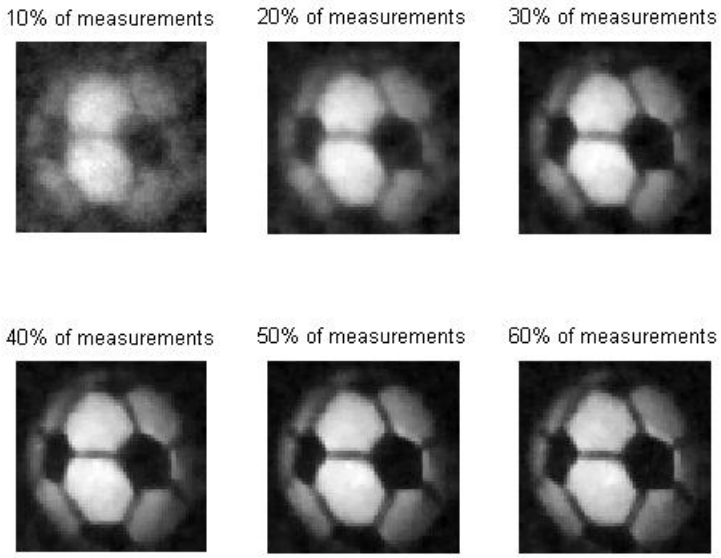

Figura 11 – Ball obtida por meio de pdNCGs.

Nesse teste a avaliação da reconstrução é feita de modo visual. É possível notar que os métodos obtiveram resultados similares quanto à qualidade da imagem. Os tempos

levados para a recuperação da imagem foram $3,5$ segundos para pdNCG, e $2,3$ segundos para pdNCGs. Ou seja, obtemos um tempo um pouco melhor.

7.4 Câmera single-pixel

No segundo teste, são comparados os métodos TFOCS_con, pdNCG e pdNCGs em problemas de reconstrução de imagens, em que os dados são amostrados usando uma câmera *single-pixel*. A solução ótima é desconhecida, como também o nível de ruído, dessa forma, a reconstrução da imagem só pode ser comparada visualmente. Para todos os experimentos, a matriz $A \in \mathbb{R}^{m \times n}$, é uma base parcial de Walsh, com $n = 64^2$ e $m \approx 0,4n$. Para todos os experimentos, as 40% de medidas são selecionadas de forma uniforme e aleatória.

As reconstruções das imagens são apresentadas nas Figuras 12, 13, 14, 15 e 16. As tabelas seguintes as imagens, constam os tempos em segundos, de cada método para a solução. Os tempos que pdNCGs obteve o melhor resultado estão destacados em negrito.

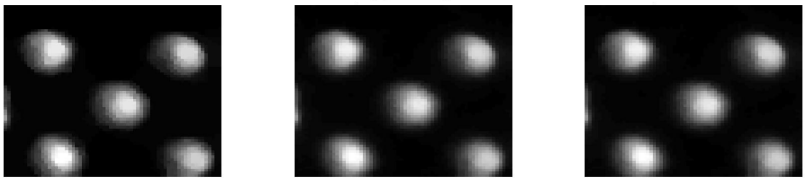

Figura 12 – Figura Dice reconstruída via TFOCS_con, pdNCG e pdNCGs, respectivamente.

TFOCS_con	**pdNCG**	**pdNCGs**
11,7	5,5	**2,4**

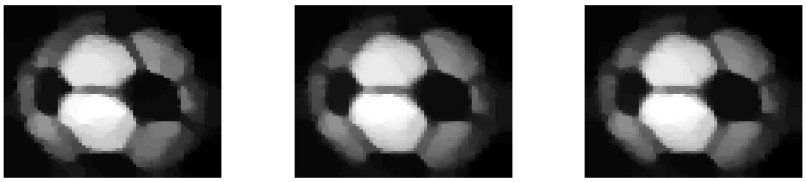

Figura 13 – Figura Ball reconstruída via TFOCS_con, pdNCG e pdNCGs, respectivamente.

TFOCS_con	**pdNCG**	**pdNCGs**
11,8	16,8	21,1

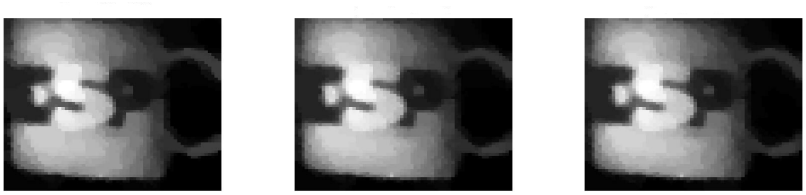

Figura 14 – Figura Cup reconstruída via TFOCS_con, pdNCG e pdNCGs, respectivamente.

TFOCS_con	**pdNCG**	**pdNCGs**
18,5	14,7	19,3

Figura 15 – Figura Letter reconstruída via TFOCS_con, pdNCG e pdNCGs, respectivamente.

TFOCS_con	**pdNCG**	**pdNCGs**
11,8	28,9	12,1

Figura 16 – Figura Logo reconstruída via TFOCS_con, pdNCG e pdNCGs, respectivamente.

TFOCS_con	**pdNCG**	**pdNCGs**
23,1	33,2	**10,8**

Notamos que, de uma forma geral, as imagens alcançam o mesmo nível de reconstrução. Os métodos pdNCGs e TFOCS obtiveram um melhor tempo em duas figuras, pdNCG em uma. No artigo de Dassios, Fountoulakis e Gondzio (2015), é comentado que fazendo certos ajustes em pdNCG, poderia ser obtido um melhor tempo em todas as imagens, mas que decidiu-se usar a implementação padrão, para tornar a comparação mais justa. De uma forma geral pdNCGs e TFOC obtiveram os melhores desempenhos, sendo que, pode-se dizer que pdNCGs foi mais eficiente, visto que na imagem Dice ele obteve um tempo quase 5 vezes melhor que TFOCs e em Logo foi aproximadamente 2 vezes mais rápido; já nas imagens Ball e Letter, em que TFOCS obteve os melhores tempos, a diferença dos tempos obtidos em Letter é pequena, e em Ball o tempo de pdNCGs, chega a 2 vezes o de TFOCS, aproximadamente.

7.5 Desempenho em relação ao nível de ruído

Nesse terceiro teste, é analisado o desempenho dos programas quando o nível de ruído no problema é aumentado (portanto, para a redução de PSNR). PSNR é medido em dB (decibéis), que é uma unidade logarítmica que indica a proporção de uma quantidade física (geralmente energia ou intensidade) em relação a um nível de referência especificado ou implícito (LAVERS, 2017). Nesse experimento as imagens utilizadas foram: House, Peppers, Lena, Fingerprint, Boat e Barbara (Figuras 6 e 7). A Tabela 7 mostra os resultados obtidos para o decrescimento de PSNR da imagem original, de 90 dB para 15 dB, realizado em seis passos. Na tabela também consta o PSNR obtido pelos métodos, quando a imagem original é corrompida por ruído, e o tempo requerido pelos métodos, dado em segundos. A matriz *Compressive Sensing* (CS) A, para todos os experimentos, é uma matriz parcial Transformada Discreta do Cosseno (DCT), com $m \approx \frac{n}{4}$.

Denotamos em negrito os valores de tempo e PSNR que nosso método pdNCGs obteve um melhor resultado que os demais métodos. O asterisco corresponde aos problemas em que não é obtida uma boa reconstrução da imagem, isso é determinado comparando o valor do PSNR obtido do método ao PSNR obtido pelo método TFOCS_unc.

Tabela 7 – Desempenho de TFOCs, TVAL3, pdNCG e pdNCGs, para o aumento de ruído. A tabela mostra o tempo em segundos e o correspondente PSNR da reconstrução da imagem.

Solver	**PSNR**	**PSNR**	**House**	**PSNR**	**Peppers**	**PSNR**	**Lena**
TFOCS_con	90	20,0	29,3	31,9	28,4	29,5	176,1
	75	20,0	28,7	31,9	29,9	29,5	178,0
	60	20,0	28,5	31,8	29,4	29,5	177,8
	45	20,0	29,4	31,6	28,3	29,5	178,8
	30	19,9	29,2	29,7	28,8	28,6	178,5
	15	19,2	28,7	24,3	28,5	25,6	178,7
TFCOS_unc	90	20,0	40,4	30,4	39,6	29,3	258,7
	75	20,0	39,9	30,4	41,6	29,3	262,3
	60	20,0	40,1	30,4	40,5	29,3	259,5
	45	20,0	40,5	30,3	39,5	29,3	260,0
	30	19,9	39,4	29,7	39,6	28,8	259,0
	15	18,2	40,4	23,1	40,0	22,6	261,5
TVAL3	90	20,0	1,5	30,7	1,9	29,3	10,9
	75	20,0	1,4	30,7	1,6	29,3	10,9
	60	20,0	1,8	30,8	1,6	29,4	11,2
	45	20,0	1,5	30,6	1,7	29,3	13,3
	30	19,9	1,3	29,7	1,7	28,7*	734,3
	15	17,7*	85,7	21,6*	91,8	21,9*	640,6
pdNCG	90	20,0	17,3	30,4	59,5	29,3	130,2
	75	20,0	17,2	30,4	77,7	29,3	131,0
	60	20,0	16,8	30,3	236,1	29,3	129,7
	45	20,0	17,2	30,3	51,5	29,3	152,6
	30	19,9	11,4	29,7	35,5	28,8	95,6
	15	18,3	13,6	23,4	13,6	22,7	85,8
pdNCGs	90	**20,4**	6,9	30,4	17,6	**29,9**	36,0
	75	**20,4**	6,2	30,4	17,5	**29,9**	35,6
	60	**20,4**	6,3	30,3	19,1	**29,9**	35,6
	45	**20,4**	6,5	30,3	18,9	**29,9**	36,5
	30	**20,3**	6,1	29,7	13,7	**29,5**	**38,0**
	15	18,7	**6,0**	23,6	**5,1**	23,1	**42,6**

Solver	**PSNR**	**PSNR**	**Fingerprint**	**PSNR**	**Boat**	**PSNR**	**Barbara**
TFOCS_con	90	20,1	178,3	27,6	178,7	25,0	178,1
	75	20,1	176,9	27,6	179,1	25,0	177,2
	60	20,1	178,2	27,6	179,6	25,0	177,5
	45	20,1	179,2	27,5	178,6	24,9	178,9
	30	19,7	177,4	26,9	177,1	24,7	179,2
	15	17,9	176,5	24,0	177,0	22,6	176,6
TFCOS_unc	90	20,0	259,7	27,5	261,1	24,9	260,2
	75	20,0	260,5	27,5	260,1	24,9	255,9
	60	20,0	260,9	27,5	261,9	24,9	259,3
	45	20,0	259,0	27,4	261,0	24,9	259,7
	30	19,9	258,9	27,0	259,0	24,7	259,7
	15	18,1	260,0	21,9	259,5	21,0	258,5
TVAL3	90	20,3	6,6	27,5	23,6	24,9	36,2
	75	20,3	6,5	27,5	25,2	24,9	35,9
	60	20,3	6,4	27,5	24,7	24,9	36,0
	45	20,3	6,8	27,4	15,8	24,9	28,4
	30	20,3	5,4	27,0	23,7	24,6*	689,4
	15	18,4	5,4	21,9	7,0	20,6*	650,6
pdNCG	90	20,0	122,6	27,5	161,6	24,9	188,2
	75	20,0	122,9	27,5	162,9	24,9	187,8
	60	20,0	122,6	27,5	160,7	24,9	188,9
	45	20,0	122,0	27,4	101,9	24,9	189,4
	30	19,9	117,9	27,0	97,2	24,7	181,2
	15	18,2	123,0	22,0	88,0	21,1	90,2
pdNCGs	90	20,0	56,3	**27,8**	39,3	24,9	137,9
	75	20,0	57,2	**27,8**	39,0	24,9	120,7
	60	20,0	55,1	**27,8**	38,1	24,9	139,3
	45	20,0	56,4	**27,8**	38,7	24,9	160,2
	30	20,0	55,5	**27,4**	37,2	24,6	**149,8**
	15	18,3	89,2	22,3	43,5	21,2	**41,7**

Como já foi mencionado em Dassios, Fountoulakis e Gondzio (2015), o método pdNCG tem bom desempenho para problemas com grande nível de ruído, o que se repete para o novo método.

Comparando nosso método pdNCGs com sua versão original, pdNCG, constatamos que obtivemos uma redução de tempo muito significativa, alcançando em alguns casos, um tempo quatro vezes menor.

7.6 Desempenho em relação ao tamanho do problema

Nesse teste analisamos o desempenho das implementações à medida que o tamanho do problema, n, aumenta. Nesse experimento a imagem *Shepp-Logan* (9) foi

utilizada. A matriz *Compressive Sensing* (CS) A é uma matriz parcial DCT (RAO; YIP, 2014), com $m \approx \frac{n}{4}$. Os sinais amostrados tem PSNR igual a 10 dB. A Tabela 8 mostra os resultados obtidos para tempo e PSNR, com o tamanho variando de 64^2 a 1024^2 pixels.

Tabela 8 – Desempenho de TwIST, TFOCS, TVAL3, pdNCG e pdNCGs para o aumento do tamanho do problema. A imagem *Shepp-Logan* foi usada para este experimento. A tabela mostra o tempo em segundos e o correspondente PSNR da reconstrução da imagem.

Solver	**n**	64^2	128^2	256^2	512^2	1024^2
TwIST	CPU tempo(s)	25,3	64,5	193,7	1651,8	6762,5
	PSNR	13,5*	15,9*	16,8*	16,9*	16,9*
TFOCS_con	CPU tempo(s)	6,7	10,8	29,6	182,6	729,1
	PSNR	15,7*	17,4	17,9	18,0	18,0
TFCOS_unc	CPU tempo(s)	7,5	13,9	42,0	265,6	1070,9
	PSNR	15,8	17,4	17,9	18,0	18,0
TVAL3	CPU tempo(s)	0,3	0,7	125,5	890,2	3455,9
	PSNR	15,8	17,5	17,2*	17,2*	17,3*
pdNCG	CPU tempo(s)	1,1	3,2	8,1	48,2	194,8
	PSNR	15,8	17,4	17,9	18,0	18,0
pdNCGs	CPU tempo(s)	0,8	3,5	8,4	51,8	213,4
	PSNR	**16,0**	17,5	17,9	18,0	18,0

Em negrito, destacamos onde o método pdNCGs obteve o melhor PSNR, notamos que isso é alcançado quando a imagem possui 64^2 pixels. Os problemas que não convergiram, foram indicados por um asterisco.

Note que TVAL3 não convergiu para uma solução de maior ou igual PSNR do método TFCOS_unc para três tamanhos de imagem (256^2, 512^2 e 1024^2). O método TwIST não alcançou a solução desejada de PSNR para a reconstrução da imagem em nenhum caso. De uma forma geral, os métodos do tipo pdNCG possuem o melhor desempenho conforme aumenta-se o tamanho do problema, isso pode ser notado pelos tempos necessários para a solução. Comparando pdNCGs com pdNCG, notamos que o novo método só ganha no tempo para a imagem de menor quantidade de pixels, apesar das diferenças de tempos entre os dois métodos não serem grandes; mas nota-se que nos dois primeiros tamanhos de imagem (64^2 e 128^2), obtemos uma melhor qualidade na reconstrução da imagem.

7.7 Desempenho em relação ao parâmetro de suavização

Nesse teste, analisamos a dependência do método em relação ao parâmetro de suavização μ. É apresentado o desempenho dos métodos pdNCGs, pdNCG e este último, sem pré-condicionador, que indicaremos na Tabela 9 por PCGs, PCG e CG respectivamente. Utilizamos para os testes as imagens: House, Peppers, Lena, Fingerprint, Boat e Barbara (Figuras 6 e 7). A matriz *Compressive Sensing* (CS) A, é uma matriz parcial DCT, com

$m \approx \frac{n}{4}$, e n igual ao número de pixels de cada imagem. Para todos os experimentos os sinais amostrados tem PSNR igual a 15 decibéis (dB). Os resultados obtidos por cada método em relação ao tempo e PSNR obtidos são apresentados na Tabela 9, respectivamente.

Tabela 9 – Desempenho dos métodos para o decrescimento do parâmetro μ. A tabela mostra o tempo em segundos e o correspondente PSNR da reconstrução da imagem.

μ		**House**	**Peppers**	**Lena**	**FingerPrint**	**Boat**	**Barbara**
	PCGs	2,2	2,2	**13,2**	**12,8**	**12,8**	**12,8**
	PSNR	**19,5**	**24,5**	**25,9**	**18,3**	**24,2**	22,6
	PCG	2,6	2,5	16,4	15,0	18,0	14,7
10^{-2}	**PSNR**	19,2	24,4	25,6	18,1	24,0	22,7
	CG	2	2,1	15,0	13,2	15,4	13,0
	PSNR	19,2	24,4	25,6	18,1	24,0	22,7
	PCGs	**4,5**	**4,1**	**27,5**	**94,6**	**27,9**	**27,7**
	PSNR	**19,6**	24,4	**25,9**	**18,2**	**24,2**	22,6
	PCG	6,0	6,8	31,3	103,7	44,5	35,1
10^{-4}	**PSNR**	19,2	24,4	25,6	18,0	24,0	22,6
	CG	5,3	6,1	57,3	107,0	57,2	38,5
	PSNR	19,2	24,4	25,6	18,0	24,0	22,6
	PCGs	**6,4**	**5,7**	**38,2**	286,5	**42,0**	112,3
	PSNR	**19,7**	24,4	**25,9**	**18,1**	**24,1**	22,6
	PCG	9,4	8,9	51,6	211,8	54,0	83,6
10^{-7}	**PSNR**	19,3	24,4	25,6	18,0	24,0	22,6
	CG	49,1	45,3	791,2	1950,7	1013,7	576,7
	PSNR	19,3	24,4	25,6	18,0	24,0	22,6
	PCGs	**9,1**	**7,7**	**60,2**	1885,4	**68,0**	277,3
	PSNR	**19,7**	24,4	**25,9**	**18,1**	**24,1**	22,6
	PCG	10,6	11,2	65,2	186,8	68,5	80,5
10^{-10}	**PSNR**	19,3	24,4	25,6	18,0	24,0	22,6
	CG	119,5	122,0	2476,2	77160,4	2047,8	2319,5
	PSNR	19,3	24,4	25,6	17,8	24,0	22,6
	PCGs	**8,4**	**6,3**	**51,6**	3726,9	**66,9**	461,3
	PSNR	**19,6**	24,4	**25,9**	**18,1**	**24,1**	22,6
	PCG	11,2	11,6	68,0	235,3	71,5	90,1
10^{-13}	**PSNR**	19,3	24,4	25,6	18,0	24,0	22,6
	CG	152,1	131,8	3285,6	82427,6	3253,7	65147,8
	PSNR	19,3	24,4	25,6	17,4	24,0	22,4

Em negrito, destacamos os problemas em que pdNCGs obteve o melhor tempo, ou a melhor qualidade de imagem. Notamos que para cada parâmetro μ, pdNCGs obteve pelo menos 4 das 6 melhores reconstruções possíveis. Para as imagens FingerPrint e Barbara, pdNCGs obteve grande tempo para a solução, quando o parâmetro μ vale 10^{-10} e 10^{-13}, isso comparado ao tempo obtido por pdNCG.

Sabemos da Seção 6.6, que o pré-condicionador só é ativado, quando o parâmetro μ é menor ou igual a 10^{-4}. Isso é constatado na tabela, em que o tempo para convergência nos métodos pré-condicionados tem um grande aumento quando muda-se do parâmetro $\mu = 10^{-2}$ para $\mu = 10^{-4}$. Quando o pré-condicionador está ativo, notamos que o tempo é estável para a variação do μ, o que evidencia a eficiência do pré-condicionador.

7.8 Desempenho em relação ao número de medidas

Nesse experimento, comparamos quatro métodos para o decrescimento do número de medidas m. As figuras utilizadas para o teste são: House, Peppers, Lena, Fingerprint, Boat e Barbara (Figuras 6 e 7).

A matriz *Compressive Sensing* (CS) A é uma matriz parcial DCT. Para todos os experimentos, os sinais amostrados tem PSNR igual a 15 dB.

Em negrito, destacamos os problemas que pdNCGs obteve os melhores tempos, comparado aos outros métodos. Note que para 75% de amostras, o que significa que $m \approx \frac{3n}{4}$, em que n é o número de pixels da imagem a ser reconstruída, pdNCGs obteve o melhor tempo em todas as comparações. Comparando pdNCGs a pdNCG foram poucos os casos que pdNCG não obteve um tempo maior que o dobro alcançado em nosso método. O método TVAL3 não obteve uma boa reconstrução da imagem, para a maioria dos problemas.

Tabela 10 – Desempenho dos métodos para o número de medidas m. A tabela mostra o tempo em segundos e o correspondente PSNR da reconstrução da imagem.

Solver	m	House	Peppers	Lena	FingerPrint	Boat	Barbara
	75%	38,2	37,8	213,9	213,3	211,0	211,6
	PSNR	19,6	27,9	27,2	19,9	25,9	24,8
	50%	36,0	36,5	207,8	206,7	205,1	204,1
TFOCS_con	PSNR	19,5	26,6	26,7	19,2	25,2	23,8
	25%	36,7	35,3	202,2	197,4	198,2	196,2
	PSNR	19,2	24,3	25,6	17,9	24,0	22,6
	75%	50,4	50,5	295,1	293,7	291,3	291,7
	PSNR	17,5	22,2	21,0	18,4	20,9	21,2
	50%	50,4	49,8	294,1	289,1	287,6	289,4
TFOCS_unc	PSNR	17,8	22,8	21,7	18,5	21,4	21,3
	25%	48,3	50,1	290,7	287,8	289,1	292,6
	PSNR	18,2	23,1	22,6	18,1	21,9	21,0
	75%	176,7	187,4	1185,0	1181,4	1026,6	931,7
	PSNR	17,0*	21,0*	20,5*	18,1*	20,4*	20,8*
	50%	146,8	154,2	5,9	4,8	5,7	6,6
TVAL3	PSNR	17,4*	21,5*	21,7	18,6	21,5	21,4
	25%	104,0	115,5	725,8	5,9	7,6	731,3
	PSNR	17,7*	21,6*	21,9*	18,4	21,9	20,6*
	75%	24,0	24,7	134,0	137,4	140,6	141,9
	PSNR	17,5	22,3	21,0	18,4	20,9	21,2
	50%	22,2	22,1	132,0	156,5	132,7	136,0
pdNCG	PSNR	17,9	23,0	21,8	18,5	21,5	21,3
	25%	16,4	15,6	99,8	141,4	102,3	102,3
	PSNR	18,3	23,4	22,7	18,2	22,0	21,1
	75%	**6,7**	**7,3**	**47,3**	**49,8**	**46,5**	**46,3**
	PSNR	17,8	22,3	21,2	18,6	21,1	21,2
	50%	**7,3**	**7,9**	52,2	80,9	53,2	53,0
pdNCGs	PSNR	18,2	23,1	22,0	18,8	21,7	21,3
	25%	**7,1**	**5,7**	**49,1**	96,6	48,6	**46,6**
	PSNR	18,7	23,6	23,1	18,3	22,3	21,2

Expostos os resultados obtidos nos seis testes em que analisamos o desempenho do pré-condicionador pdNCGs aplicado a problemas *Compressive Sensing*, na seção seguinte apresentamos nossa segunda abordagem.

7.9 Segunda Abordagem

Nessa seção apresentamos dois métodos desenvolvidos para o problema *Compressive Sensing*. No primeiro método modificações no cálculo das direções primal-dual são realizadas. No segundo, além das condições de otimalidade em relação à variável primal, também consideramos as condições em relação à variável dual.

7.9.1 Primeiro Método

Na análise do desenvolvimento do método pdNCG verificamos as direções do método primal-dual serão obtidas resolvendo o seguinte sistema linear:

$$\begin{pmatrix} A^TA & \tau ReW & \tau ImW \\ -B_2ImW^T + B_1ReW^T - ReW^T & D^{-1} & 0 \\ -B_3ReW^T + B_4ImW^T - ImW^T & 0 & D^{-1} \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta g_{re} \\ \Delta g_{im} \end{pmatrix} = \begin{pmatrix} -\nabla f_\tau^\mu(x) \\ -D^{-1}g_{re} + ReW^Tx \\ -D^{-1}g_{im} + ImW^Tx \end{pmatrix},$$

que corresponde a:

$$\begin{aligned} &A^TA\Delta x + \tau ReW\Delta g_{re} + \tau ImW\Delta g_{im} = -\nabla f_\tau^\mu(x) = r_1 \\ &((B_1 - I)ReW^T - B_2ImW^T)\Delta x + D^{-1}\Delta g_{re} = -D^{-1}g_{re} + ReW^Tx = r_2 \\ &((B_4 - I)ImW^T - B_3ReW^T)\Delta x + D^{-1}\Delta g_{im} = -D^{-1}g_{im} + ImW^Tx = r_3 \end{aligned} \quad . \tag{7.2}$$

Da segunda e terceira equações, temos:

$$\begin{aligned} \Delta g_{re} &= D(I - B_1)ReW^T\Delta x + DB_2ImW^T\Delta x - g_{re} + DReW^Tx \\ \Delta g_{im} &= D(I - B_4)ImW^T\Delta x + DB_3ReW^T\Delta x - g_{im} + DImW^Tx \end{aligned} \quad . \tag{7.3}$$

E fazendo a substituição na primeira equação de (7.2), obtemos:

$$\begin{aligned} &A^TA\Delta x + \tau(ReW(D(I - B_1)ReW^T + DB_2ImW^T)\Delta x) + \tau(ImW(D(I - B_4)ImW^T \\ &+DB_3ReW^T)\Delta x) + \tau ReW(DReW^Tx - g_{re}) + \tau ImW(DImW^Tx - g_{im}) = -\nabla f_\tau^\mu(x). \end{aligned} \tag{7.4}$$

Mas notamos que nessa substituição, Dassios, Fountoulakis e Gondzio (2015) consideram:

$$\tau ReW(DReW^Tx - g_{re}) + \tau ImW(DImW^Tx - g_{im}) = 0.$$

Assim o sistema a ser resolvido é:

$$\begin{aligned} \hat{B}\Delta x &= -\nabla f_\tau^\mu(x) \\ \Delta g_{re} &= D(I - B_1)ReW^T\Delta x + DB_2ImW^T\Delta x - g_{re} + DReW^Tx \\ \Delta g_{im} &= D(I - B_4)ImW^T\Delta x + DB_3ReW^T\Delta x - g_{im} + DImW^Tx \end{aligned} \quad , \tag{7.5}$$

ao invés de:

$$\begin{aligned} \hat{B}\Delta x &= -\nabla f_\tau^\mu(x) - \tau ReW(DReW^Tx - g_{re}) - \tau ImW(DImW^Tx - g_{im}) \\ \Delta g_{re} &= D(I - B_1)ReW^T\Delta x + DB_2ImW^T\Delta x - g_{re} + DReW^Tx \\ \Delta g_{im} &= D(I - B_4)ImW^T\Delta x + DB_3ReW^T\Delta x - g_{im} + DImW^Tx \end{aligned} \quad . \tag{7.6}$$

Considerando que com a eliminação desses dados, o método pode vir a ser menos eficiente, realizamos alguns testes e constatamos que considerando na primeira linha de (7.6) apenas os resíduos referentes a variável dual g_{re} (r_2), ou seja:

$$\hat{B}\Delta x = -\nabla f_\tau^\mu(x) - \tau ReW(DReW^Tx - g_{re}),$$

resultados melhores ao que pdNCG obteve podem ser alcançados.

7.9.2 Segundo Método

No Capítulo 6 foram apresentadas a formulação primal-dual (6.8) e as condições de otimalidade (6.10) consideradas no método pdNCG. Na tentativa de obter resultados computacionais melhores em nosso segundo método, consideramos a condição de otimalidade em relação as variáveis duais g_{re} e g_{im}, desconsiderando a segunda e terceira equação de (6.10).

Calculando o gradiente de (6.8) em relação a g_{re} e g_{im}, e igualando a zero, obtemos:

$$\begin{aligned} Re(W^T)x &= \sum_{i=1}^{l} \mu \frac{Re(e_i g_i)}{(1-\|g_i\|^2)^{\frac{1}{2}}} \\ Im(W^T)x &= \sum_{i=1}^{l} \mu \frac{Im(e_i g_i)}{(1-\|g_i\|^2)^{\frac{1}{2}}} \end{aligned} \quad . \tag{7.7}$$

Portanto, nossas condições de otimalidade são:

$$\begin{aligned} \tau Re(W\overline{g}) + A^T(Ax-b) &= 0 \\ Re(W^T)x &= \sum_{i=1}^{l} \mu \frac{Re(e_i g_i)}{(1-\|g_i\|^2)^{\frac{1}{2}}} \\ Im(W^T)x &= \sum_{i=1}^{l} \mu \frac{Im(e_i g_i)}{(1-\|g_i\|^2)^{\frac{1}{2}}} \end{aligned} \quad . \tag{7.8}$$

Para o cálculo das direções primal-dual, resolvemos o sistema linear:

$$\begin{pmatrix} A^TA & \tau ReW & \tau ImW \\ -ReW^T & \blacksquare & \blacktriangle \\ -ImW^T & \blacktriangle & \bullet \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta g_{re} \\ \Delta g_{im} \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix}, \tag{7.9}$$

sendo que,

$$\bullet = \sum_{i=1}^{l} \mu \frac{e_i e_i^T}{(1-\|g_i\|^2)^{\frac{1}{2}}} + \mu \frac{(Img_i)^2 e_i e_i^T}{(1-\|g_i\|^2)^{\frac{3}{2}}},$$

$$\blacksquare = \sum_{i=1}^{l} \mu \frac{e_i e_i^T}{(1-\|g_i\|^2)^{\frac{1}{2}}} + \mu \frac{(Reg_i)^2 e_i e_i^T}{(1-\|g_i\|^2)^{\frac{3}{2}}},$$

$$\blacktriangle = \sum_{i=1}^{l} \mu \frac{Re(g_i)Im(g_i) e_i e_i^T}{(1-\|g_i\|^2)^{\frac{3}{2}}},$$

$$r_1 = -\nabla f_c^{\mu},$$

$$r_2 = -\sum_{i=1}^{l} \mu \frac{Re(e_i g_i)}{(1-\|g_i\|^2)^{\frac{1}{2}}} + ReW^T x,$$

$$r_3 = -\sum_{i=1}^{l} \mu \frac{Im(e_i g_i)}{(1-\|g_i\|^2)^{\frac{1}{2}}} + ImW^T x.$$

Note que as matrizes ●, ■ e ▲ são diagonais. Da segunda e terceira equação de (7.9.2), obtemos:

$$\Delta g_{im} = (1 - ●^{-1}▲■^{-1}▲)^{-1}(●^{-1}r_3 - ●^{-1}▲■^{-1}r_2 + ●^{-1}(ImW^T\Delta x - ▲■^{-1}ReW^T\Delta x)),$$
$$\Delta g_{re} = ■^{-1}(r_2 + ReW^T\Delta x - ▲\,\Delta g_i).$$

Substituindo Δg_{im} e Δg_{re} na primeira equação de (7.9.2), e fazendo $r_2 = r_3 = 0$, resolvemos a seguinte equação para determinar Δx:

$$\begin{aligned}&(AA^T + (\tau ReW(■^{-1} + ■^{-1}▲\,(1 - ●^{-1}▲■^{-1}▲)^{-1}●^{-1}▲■^{-1})ReW^T\\ &-\tau ReW(■^{-1}▲\,(1 - ●^{-1}▲■^{-1}▲)^{-1}●^{-1})ImW^T - \tau ImW((1 - ●^{-1}▲■^{-1}▲)^{-1}\\ &(●^{-1}▲■^{-1}))ReW^T + \tau ImW((1 - ●^{-1}▲■^{-1}▲)^{-1}●^{-1})ImW^T))\Delta x = -\nabla f_c^{\mu}.\end{aligned} \tag{7.10}$$

Portanto, em nossos testes utilizamos a implementação de Fountoulakis (2015) fazendo as mudanças adequadas para o cálculo das direções e resíduos.

Análise do Método

Em Fountoulakis (2015), o pré-condicionador é desenvolvido tendo em vista que as componentes de $W^*x_{\tau,\mu}$ variam entre os dois conjuntos disjuntos:

$$\mathcal{B} := \{i \in \{1,2,...,l\} \mid |W_i^* x_{\tau,\mu}| >> \mathcal{O}(w)\}, \quad |\mathcal{B}| = q = |sup(W^* x_\tau)|,$$

$$\mathcal{B}^c := \{i \in \{1,2,...,l\} \mid |W_i^* x_{\tau,\mu}| \approx \mathcal{O}(w)\}, \quad |\mathcal{B}^c| = l - q,$$

em que $sup(u) = \{i \in \{1,2,...,l\} \mid |u_i| \neq 0\}$, e c denota o complemento do conjunto. Assim, as componentes da matriz diagonal D, definida em (6.4), variam em dois conjuntos distintos. Em particular, q componentes são muito menores que $\mathcal{O}(\frac{1}{w})$, enquanto a maioria dos componentes $(l-q)$, são da ordem $\mathcal{O}\left(\frac{1}{w}\right)$, ou seja,

$$D_i << \mathcal{O}\left(\frac{1}{w}\right), \quad \forall i \in \mathcal{B} \quad \text{e} \quad D_i = \mathcal{O}\left(\frac{1}{w}\right), \quad \forall i \in \mathcal{B}^c,$$

sabendo que w é definido na Seção 5.4 de Fountoulakis (2015), como um constante arbitrária positiva, tal que existe um parâmetro suficientemente pequeno μ com $\|x_{\tau,\mu} - x_\tau\|_2 < w$. Dessa forma, a matriz $\nabla^2 f_\tau^\mu$ em (6.7), contém uma matriz dominante $\nabla^2\psi_\mu$, e a matriz A^TA com um maior autovalor moderado, pois temos que $\|AA^T - I_m\| \leqslant \delta$ de (4.23), para alguma constante $\delta \geqslant 0$ de módulo pequeno, e $\lambda_{max}(A^TA) = \lambda_{max}(AA^T) \Rightarrow \lambda_{max}(A^TA) \leqslant 1 + \delta$.

Sabendo que a matriz simétrica mais próxima de $\widetilde{B}$ de acordo com a norma de Frobenius, sendo $\widetilde{B}$ definido em (6.14), tende para a matriz $\nabla^2\psi_\mu$ quando $x \to x_{\tau,\mu}$, a matriz $sym(\widetilde{B})$ é a matriz dominante em $\widehat{B}$, definido em (6.16). Dessa forma, no pré-condicionador proposto, a matriz A^TA é substituída pela matriz ρI_n, $\rho > 0$, enquanto a matriz dominante $sym(\widetilde{B})$ é mantida. O pré-condicionador proposto é dado por:

$$\widetilde{N} := c\; sym(\widetilde{B}) + \rho I_n.$$

E com o objetivo de limitar os elementos da matriz diagonal D, para pontos próximos a $x_{\tau,\mu}$, quando μ é suficientemente pequeno, os conjuntos $\mathcal{B}$ e $\mathcal{B}^c$ serão aproximados. Define-se uma constante ν, tal que $D_i < \nu$. Dessa forma, temos:

$$\mathcal{B}_\nu := \{i \in \{1,2,...,l\} \mid D_i < \nu\}, \quad e \quad \mathcal{B}^c_\nu := \{1,2,...,l\} \setminus \mathcal{B}_\nu,$$

com $|\mathcal{B}_\nu| = \sigma$, e $|\mathcal{B}^c_\nu| = l - \sigma$. Podendo σ ser diferente de q, esparsidade de $W^* x_\tau$.

Em nossa implementação o sistema obtido é bem diferente do anterior, isso pode ser verificado comparando as equações (7.10) e a primeira linha do sistema (6.15). Podemos ver por (7.10), que obtemos matrizes diagonais, cujos elementos possuem no denominador o termo $(1 - \|g_i\|^2)$ elevado a um certo expoente. Nos experimentos numéricos, este termo $(1 - \|g_i\|^2)$ mostrou-se próximo a zero, sendo assim, acrescentamos uma perturbação. Como temos que $\|g\|_\infty \leqslant 1$, imaginávamos que tal problema pudesse ocorrer.

Note que, fazendo um paralelo ao sistema (6.15) analisado anteriormente, com o nosso; no de Fountonlaskis, o termo em análise é a matriz D, sendo visto que possui elementos em dois conjuntos distintos. Tal matriz multiplica todos os termos das somas que definem $\widetilde{B}$. Em nosso caso temos a matriz diagonal $\mathcal{D}$, com elementos iguais a $\dfrac{(1 - \|g_i\|^2)^2}{\mu(1 - \|g_i\|^2)^{\frac{3}{2}}} = \dfrac{(1 - \|g_i\|^2)^{\frac{1}{2}}}{\mu}$ multiplicando todos os termos. Veja que para $(1 - \|g_i\|^2)$ próximo ao valor de μ (lembrando que quanto menor o valor de μ, melhor é nossa aproximação da função f^μ_τ a f_τ), obtemos um elemento com valor absoluto não muito grande. Agora, para $(1 - \|g_i\|^2)$ com um valor não próximo a zero, obtemos elementos de valor absoluto grande. Por exemplo, se $(1 - \|g_i\|^2) = 0,5$, para $\mu = 10^{-5}$, obtemos um elemento igual a 50000. Dessa forma, também ocorrem termos na diagonal com valor absoluto grande e pequeno. Em nossos testes, o pré-condicionador elaborado por Fountoulakis (2015) obteve bons resultados. Acreditamos que seja pelo motivo citado anteriormente.

Constatamos que diferente da implementação introduzida no Capítulo 6, em que o pré-condicionador é utilizado apenas nas últimas iterações, foi necessário o uso do mesmo em todas elas; uma explicação possível seria por nossa implementação ter o termo $(1 - \|g_i\|^2)$ próximo a zero, o que torna a implementação instável, e a de Fountoulakis (2015) não possuir esse problema, podendo o mal condicionamento da matriz ser tratada apenas pela variação dos parâmetros μ e τ, por meio do Método da Continuação.

7.9.3 Experimentos Numéricos

Apresentamos os resultados obtidos para os métodos expostos na segunda abordagem 7.9.1, assim também como a implementação de pdNCG e dois métodos de primeira ordem de estado da arte: *Templates for First-Order Conic Solvers* (TFOCS (BECKER; CANDÈS; GRANT, 2011)) e *Total-Variation minimization by Augmented*

Lagrangian and ALternating direction ALgorithms (TVAL3 (LI et al., 2013)). Usamos a sigla pdNCGr para nossa primeira implementação e pdNCGg para a segunda.

Os experimentos numéricos foram implementados em Matlab R2014a, e realizados em um sistema operacional Microsoft Windows 10 com Intel® Core(TM) i7-5500U 2.40GHz e 8 GB de memória RAM.

As imagens utilizadas para os testes são: Ball (Figura 4), House e Peppers (Figura 6), Lena e Fingerprint (Figura 7), Boat e Barbara (Figura 8), e Shepp-Logan (Figura 9), que foram apresentadas no Seção 7.2.

Desempenho em relação ao número de medidas

Para verificar se nossa implementação consegue recuperar uma imagem de forma eficiente, nosso primeiro teste foi para uma imagem de 64^2 pixels. Usamos a demonstração da Figura 4 para isso. A seguir as imagens recuperadas para pdNCG e pdNCGg.

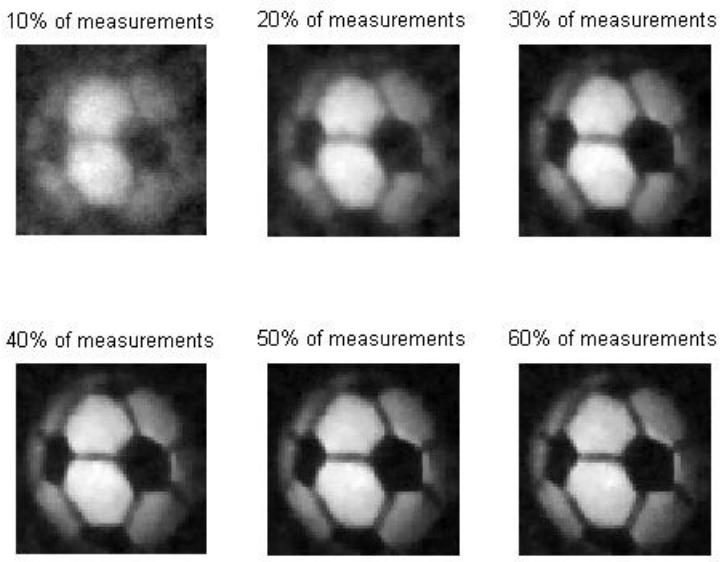

Figura 17 – Ball obtida por meio de pdNCG.

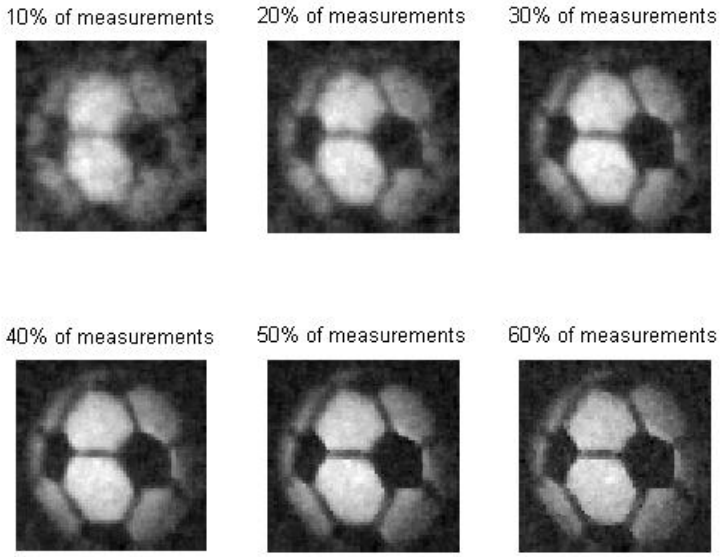

Figura 18 – Ball obtida por meio de pdNCGg.

É possível notar que pdNCG obteve uma melhor resolução que nosso método, as partes escuras da bola estão mais bem definidas. Os tempos levados para a recuperação da imagem foram $3,49$ segundos para pdNCG, e $15,06$ segundos para pdNCGg.

Desempenho em relação ao tamanho do problema

No segundo teste analisamos o desempenho das implementações em relação ao tamanho do problema. Os sinais amostrados tem PSNR igual a 15 dB. Nesse experimento a imagem *Shepp-Logan* (9) foi utilizada. A Tabela 8 mostra os resultados obtidos, com o tamanho variando de 64^2 a 1024^2 pixels.

Tabela 11 – Desempenho de TFOCS, TVAL3, pdNCG, pdNCGr e pdNCGg para o aumento do tamanho do problema. A imagem *Shepp-Logan* foi usada para este experimento. A tabela mostra o tempo em segundos e o correspondente PSNR da reconstrução da imagem.

Solver	**n**	64^2	128^2	256^2	512^2	1024^2
TFOCS_con	CPU time(s)	6,4	10,7	28,3	165,1	665,4
	PSNR	15,7	17,4	17,9	18,0	18,0
TFCOS_unc	CPU time(s)	7,5	14,0	38,9	239,2	973,7
	PSNR	15,8	17,4	17,9	18,0	18,0
TVAL3	CPU time(s)	0,3	0,8	115,8	800,8	3128,4
	PSNR	15,8	17,5	17,2	17,2	17,3
pdNCG	CPU time(s)	1,1	3,4	8,0	45,4	182,6
	PSNR	15,8	17,4	17,9	18,0	18,0
pdNCGr	CPU time(s)	1,1	3,2	7,5	44,5	181,0
	PSNR	15,8	17,4	17,9	18,0	18,0
pdNCGg	CPU time(s)	8,1	26,2	105,1	542,2	2102,5
	PSNR	15,7	17,7	18,1	17,8	17,8

O número máximo de iterações do Método de Gradientes Conjugados atingido, foi de treze iterações.

No segundo teste podemos notar que a implementação pdNCG continua mais eficiente que a nossa pdNCGg; e que pdNCGr possui resultados similares a pdNCG. Mesmo assim, alcançamos um melhor desempenho que TVAL3, obtendo uma melhor resolução, e um tempo menor, ao passo que o tamanho do problema aumenta.

Desempenho em relação ao nível de ruído

Nesse terceiro teste é analisado o desempenho dos programas quando o nível de ruído no problema é aumentado (portanto, para a diminuição de PSNR). PSNR é medido em dB. Nesse experimento as imagens utilizadas foram: House, Peppers, Lena, Fingerprint, Boat e Barbara (Figuras 6 e 7). A Tabela 7 mostra os resultados obtidos para o decrescimento de PSNR de 90 dB para 15 dB, realizado em seis passos.

Infelizmente nossa implementação pdNCGg não obteve bons resultados reconstruindo as imagens. A implementação pdNCGr alcançou uma resolução um pouco melhor da obtida por pdNCG, na Imagem Peppers, para 15 e 60 PSNR. Notamos também que para PSNR igual a 60, na Imagem Peppers, conseguimos um tempo bem menor ao alcançado por pdNCG, na implementação pdNCGr. Os últimos resultados comentados, encontram-se em vermelho na Tabela 12.

Tabela 12 – Desempenho de TFOCS, TVAL3, pdNCG, pdNCGr e pdNCGg, para o aumento de ruído. A tabela mostra o tempo em segundos e o correspondente PSNR da reconstrução da imagem.

Solver	**PSNR**	**PSNR**	**House**	**PSNR**	**Peppers**	**PSNR**	**Lena**
TFOCS_con	90	20,0	29,3	31,9	31,4	29,5	185,8
	75	20,0	30,0	31,9	29,8	29,5	175,1
	60	20,0	30,2	31,8	35,6	29,5	174,0
	45	20,0	29,9	31,6	30,0	29,5	182,7
	30	19,9	29,4	29,7	29,2	28,6	172,8
	15	19,2	28,3	24,3	29,1	25,6	170,0
TFCOS_unc	90	20,0	40,8	30,4	50,4	29,3	255,1
	75	20,0	43,6	30,4	42,9	29,3	255,7
	60	20,0	41,3	30,4	41,7	29,3	255,7
	45	20,0	39,9	30,3	56,6	29,3	250,6
	30	19,9	40,4	29,7	41,4	28,8	249,8
	15	18,2	40,5	23,1	51,2	22,6	250,5
TVAL3	90	20,0	1,4	30,7	1,7	29,3	10,5
	75	20,0	2,0	30,7	2,4	29,3	10,4
	60	20,0	1,4	30,8	1,8	29,4	11,1
	45	20,0	1,6	30,6	1,6	29,3	12,9
	30	19,9	1,3	29,7	1,6	28,7	706,5
	15	17,7	86,6	21,6	127,2	21,9	613,3
pdNCG	90	20,0	17,0	30,4	59,6	29.3	125,7
	75	20,0	17,1	30,4	86,3	29.3	126,3
	60	20,0	16,9	30,3	234,8	29.3	127,5
	45	20,0	16,1	30,3	50,3	29.3	149,0
	30	19,9	11,1	29,7	34,9	28.8	95,5
	15	18,3	14,0	23,4	14,1	22.7	82,1
pdNCGr	90	20,0	17,0	30,4	51,5	29,3	126,5
	75	20,0	17,4	30,4	56,3	29,3	124,9
	60	20,0	17,1	30,4	59,2	29,3	126,3
	45	20,0	16,1	30,3	41,3	29,3	123,5
	30	19,9	11,0	29,7	39,4	28,8	95,9
	15	18,3	14,9	23,5	10,1	22,7	90,3
pdNCGg	90	14,2	122,0	14,8	198,2	15,3	554,2
	75	14,2	117,4	14,8	228,9	15,3	556,0
	60	14,2	128,4	14,8	196,2	15,3	558,4
	45	14,2	124,7	14,8	210,6	15,3	575,8
	30	14,2	115,8	14,8	203,1	15,3	555,2
	15	13,9	122,2	14,5	174,3	14,9	585,8

Solver	**PSNR**	**PSNR**	**Fingerprint**	**PSNR**	**Boat**	**PSNR**	**Barbara**
TFOCS_con	90	20,1	183,3	27,6	172,0	25,0	170,5
	75	20,1	192,4	27,6	169,2	25,0	171,7
	60	20,1	175,1	27,6	169,6	25,0	169,3
	45	20,1	177,4	27,5	173,2	24,9	169,4
	30	19,7	173,5	26,9	171,0	24,7	169,7
	15	17,9	174,4	24,0	175,8	22,6	169,3
TFCOS_unc	90	20,0	263,7	27,5	247,2	24,9	247,7
	75	20,0	252,7	27,5	246,9	24,9	251,3
	60	20,0	258,0	27,5	252,8	24,9	245,6
	45	20,0	250,8	27,4	253,0	24,9	245,1
	30	19,9	253,8	27,0	247,4	24,7	245,7
	15	18,1	250,9	21,9	247,3	21,0	244,0
TVAL3	90	20,3	6,3	27,5	23,4	24,9	35,8
	75	20,3	6,3	27,5	23,3	24,9	35,3
	60	20,3	10,7	27,5	23,4	24,9	34,3
	45	20,3	6,9	27,4	16,2	24,9	29,2
	30	20,3	5,2	27,0	22,7	24,6	660,9
	15	18,4	5,1	21,9	6,5	20,6	614,7
pdNCG	90	20,0	138,6	27,5	156,2	24,9	181,7
	75	20,0	125,0	27,5	154,0	24,9	183,0
	60	20,0	133,5	27,5	158,2	24,9	180,4
	45	20,0	126,0	27,4	100,8	24,9	178,8
	30	19,9	115,9	27,0	93,4	24,7	175,0
	15	18,2	121,0	22,0	85,2	21,1	85,2
pdNCGr	90	20,0	119,8	27,5	129,9	24,9	132,0
	75	20,0	119,0	27,5	131,9	24,9	132,4
	60	20,0	117,2	27,5	131,4	24,9	136,9
	45	20,0	143,8	27,4	99,8	24,9	159,4
	30	19,9	115,9	27,0	64,5	24,7	154,6
	15	18,2	88,0	22,0	87,4	21,1	88,3
pdNCGg	90	14,4	566,1	15,8	549,2	14,3	605,9
	75	14,4	574,3	15,8	567,8	14,3	586,9
	60	14,4	581,6	15,8	611,9	14,3	550,6
	45	14,4	604,1	15,8	605,0	14,3	640,8
	30	14,4	588,4	15,8	613,8	14,2	597,6
	15	14,1	1838,7	15,4	561,7	14,0	1830,6

Apresentados os resultados obtidos nos três testes em que analisamos o desempenho dos pré-condicionadores pdNCGr e pdNCGg aplicados a problemas *Compressive Sensing*, no capítulo seguinte apresentamos nossas conclusões e perspectivas futuras.

Conclusões e Perspectivas Futuras

Conclusões

Neste trabalho apresentamos novos pré-condicionadores para problemas de programação linear, assim como um novo pré-condicionador para problemas *Compressive Sensing*.

Como no método de pontos interiores, obtém-se um sistema linear correspondente ao método de Newton aplicado às condições de otimalidade do problema, destacamos a importância do pré-condicionamento quando usamos de métodos iterativos para a resolução desse sistema. Destacamos alguns tipos de pré-condicionadores, dando enfoque para os pré-condicionadores Fatoração Incompleta de Cholesky e Separador; pois estes são utilizados no desenvolvimento do pré-condicionador proposto.

O novo pré-condicionador, denominado Fator Separador é apresentado, sendo que as características dos pré-condicionadores implementados nos testes computacionais são discutidas.

No primeiro experimento realizado em Matlab, foram considerados 32 problemas de programação linear. Constatamos que o novo pré-condicionador resolveu todos eles, sendo que o método que utiliza o pré-condicionador Separador em todas as iterações conseguiu resolver 29 problemas. Em relação ao tempo, conseguimos um melhor tempo em 18 problemas, sendo que para dois em especial as diferenças foram grandes. Conseguimos diminuir o número de iterações do método de pontos interiores para 5 problemas, sendo que o mesmo número de iterações foi alcançado na maioria, 18 no total.

No segundo experimento, realizado em C/Fortran, foram considerados 34 problemas das coleções: Kennington, Netlib, Mészáros, PDS e Fome. O novo método resolveu 29 e PCx_Mod 28 problemas. Na maioria dos problemas PCx_Mod obteve o menor tempo, podendo associar o maior tempo e maior número de iterações do novo método ao número de iterações realizadas no método dos gradientes conjugados. Obtivemos alguns resultados semelhantes em relação ao tempo nos dois métodos, apesar de alcançarmos um número bem maior de iterações no método dos gradientes conjugados. Na família de problemas FOME, conseguimos resolver todos os problemas, alcançando os melhores

resultados comparado ao PCx_Mod. É apresentado o perfil de desempenho em relação ao tempo, por meio dele podemos dizer que o método PCx_Mod é mais eficiente, resolvendo cerca de 64% dos problemas teste em tempo total reduzido, enquanto o método com o pré-condicionador Fator Separador resolve aproximadamente 33%. Quanto à robustez, temos que PCx_New resolve mais problemas.

Na segunda parte do trabalho, apresenta-se a teoria acerca de *Compressive Sensing*. Destacamos as condições que as matrizes do problema devem satisfazer a fim de que a solução esparsa possa ser determinada, destacamos a propriedade da RIP. Como a função objetivo do problema de interesse não é diferenciável por conter uma norma l_1, a formulação do problema por meio da função pseudo-Huber é apresentada. Um método conhecido da literatura é discutido. Tal método possui um pré-condicionador que é mais eficiente quando o método aproxima-se da solução, ou seja, seu comportamento é similar ao pré-condicionador Separador.

O novo pré-condicionador, denominado Pseudo Fator Separador, é incorporado no método comentado anteriormente, e o aplicamos nos problemas *Compressive Sensing*. Testes numéricos são realizados, em que analisamos o desempenho em relação ao número de medidas, ao nível de ruído, tamanho do problema, parâmetro de suavização, e com a câmera *single-pixel*.

De uma forma geral, em todos os testes, obtivemos resultados satisfatórios em relação ao tempo e qualidade da imagem reconstruída. Os resultados foram comparados com métodos de primeira ordem de estado da arte, e o método que utiliza o pré-condicionador que o novo método tem como base. Destacamos que para o teste com a câmera *single-pixel*, de uma forma geral o novo método e TFOC obtiveram os melhores desempenhos, sendo que pode-se dizer que pdNCGs foi mais eficiente, visto que na imagem Dice ele obteve um tempo quase 5 vezes melhor que TFOCs e em Logo, foi aproximadamente 2 vezes mais rápido. Quando analisamos o desempenho em relação ao nível de ruído, nosso método pdNCGs comparado com sua versão original, pdNCG, obteve uma redução de tempo muito significativa, alcançando em alguns casos, um tempo quatro vezes menor. E para o teste de desempenho em relação ao número de medidas, notamos que para 75% de amostras, o que significa que $m \approx \frac{3n}{4}$, em que n é o número de pixels da imagem a ser reconstruída, pdNCGs obteve o melhor tempo em todas as comparações. Ainda nesse último teste, comparando pdNCGs a pdNCG, foram poucos os casos que pdNCG não obteve um tempo maior que o dobro alcançado em nosso método, sendo que o método TVAL3 não obteve uma boa reconstrução da imagem, para a maioria dos problemas. Com base no que foi exposto, podemos dizer que o método, com o novo pré-condicionador é mais eficiente.

Perspectivas Futuras

Como trabalhos futuros pretendemos realizar uma melhor calibração dos parâmetros do novo pré-condicionador aplicado a problemas de grande porte em C e Fortran. Uma delas origina-se do fato que, devido ao número de iterações do método dos gradientes conjugados obtido pelo novo método ter alcançado um valor grande, aumentar o preenchimento do fator $\tilde{L}$ (ou seja, aumentar o valor do parâmetro η) parece uma boa ideia para melhorarmos o condicionamento da matriz. Assim, um estudo mais detalhado com o objetivo de determinar um η ótimo é pretendido. Também pretendemos realizar a Fatoração de Cholesky na matriz do pré-condicionador Separador, verificando se esse novo tipo de pré-condicionador Fator Separador é viável.

Além desses trabalhos mais focados em problemas de programação linear gerais, gostaríamos de aplicar o novo pré-condicionador a outros tipos de problemas, assim como foi feito neste trabalho para o problema *Compressive Sensing*.

Referências

ALLGOWER, E. L.; GEORG, K. *Introduction to numerical continuation methods*. Philadelphia, PA: SIAM, 2003. v. 45.

BAZARAA, M. S.; JARVIS, J. J.; SHERALI, H. D. *Linear programming and network flows*. Hoboken, NJ: John Wiley & Sons, 2011.

BAZARAA, M. S.; SHERALI, H. D.; SHETTY, C. M. *Nonlinear programming: theory and algorithms*. Hoboken, NJ: John Wiley & Sons, 2013.

BECKER, S. R.; CANDÈS, E. J.; GRANT, M. C. Templates for convex cone problems with applications to sparse signal recovery. *Mathematical programming computation*, Springer, v. 3, n. 3, p. 165–218, 2011.

BENZI, M. Preconditioning techniques for large linear systems: a survey. *Journal of computational Physics*, Elsevier, v. 182, n. 2, p. 418–477, 2002.

BERTSEKAS, D. P.; NEDIC, A.; OZDAGLAR, A. E. *Convex analysis and optimization*. Belmont, MA: Athena Scientific, 2003.

BIOUCAS-DIAS, J. M.; FIGUEIREDO, M. A. A new TwIST: Two-step iterative shrinkage/thresholding algorithms for image restoration. *IEEE Transactions on Image processing*, IEEE, v. 16, n. 12, p. 2992–3004, 2007.

BLANCHARD, J. D.; CARTIS, C.; TANNER, J. Compressed sensing: How sharp is the restricted isometry property? *SIAM review*, SIAM, v. 53, n. 1, p. 105–125, 2011.

BOCANEGRA, S.; CAMPOS, F.; OLIVEIRA, A. R. Using a hybrid preconditioner for solving large-scale linear systems arising from interior point methods. *Computational Optimization and Applications*, Springer, v. 36, n. 2, p. 149–164, 2007.

BOYD, S.; VANDENBERGHE, L. *Convex optimization*. Cambridge, UK: Cambridge university press, 2004.

CAMPOS, F. F. *Analysis of conjugate gradients-type methods for solving linear equations*. Tese (Doutorado) — University of Oxford, 1995.

CAMPOS, F. F.; BIRKETT, N. R. An Efficient Solver for Multi–Right-Hand-Side Linear Systems Based on the CCCG (η) Method with Applications to Implicit Time-Dependent Partial Differential Equations. *SIAM Journal on Scientific Computing*, SIAM, v. 19, n. 1, p. 126–138, 1998.

CANDÈS, E.; ROMBERG, J. Sparsity and incoherence in compressive sampling. *Inverse problems*, IOP Publishing, v. 23, n. 3, p. 969, 2007.

CANDÈS, E. J. Compressive sampling. In: PROCEEDINGS OF THE INTERNATIONAL CONGRESS OF MATHEMATICIANS. *Proceedings of the international congress of mathematicians*. Madrid, Spain, 2006. v. 3, p. 1433–1452.

_____. The restricted isometry property and its implications for compressed sensing. *Comptes Rendus Mathematique*, Elsevier, v. 346, n. 9, p. 589–592, 2008.

CANDÈS, E. J.; ELDAR, Y. C.; NEEDELL, D.; RANDALL, P. Compressed sensing with coherent and redundant dictionaries. *Applied and Computational Harmonic Analysis*, Elsevier, v. 31, n. 1, p. 59–73, 2011.

CANDÈS, E. J.; ROMBERG, J. K.; TAO, T. Stable signal recovery from incomplete and inaccurate measurements. *Communications on pure and applied mathematics*, Wiley Online Library, v. 59, n. 8, p. 1207–1223, 2006.

CANDÈS, E. J.; TAO, T. Decoding by linear programming. *Information Theory, IEEE Transactions on*, IEEE, v. 51, n. 12, p. 4203–4215, 2005.

CANDÈS, E. J.; WAKIN, M. B. An introduction to compressive sampling. *Signal Processing Magazine, IEEE*, IEEE, v. 25, n. 2, p. 21–30, 2008.

CESARI, L. Sulla risoluzione dei sistemi di equazioni lineari per approssimazioni successive. *Atti Accad. Naz. Lincei. Rend. Cl. Sci. Fis. Mat. Nat.*, v. 25, n. 6a, p. 422–428, 1937.

CHAN, R.; CHAN, T.; ZHOU, H. Advanced signal processing algorithms. *Proceedings of the International Society of Photo-Optical Instrumentation Engineers, FT Luk, ed., SPIE*, p. 314–325, 1995.

CHAN, T. F.; GOLUB, G. H.; MULET, P. A nonlinear primal-dual method for total variation-based image restoration. *SIAM journal on scientific computing*, SIAM, v. 20, n. 6, p. 1964–1977, 1999.

CHEN, S. S.; DONOHO, D. L.; SAUNDERS, M. A. Atomic decomposition by basis pursuit. *SIAM review*, SIAM, v. 43, n. 1, p. 129–159, 2001.

CZYZYK, J.; MEHROTRA, S.; WAGNER, M.; WRIGHT, S. J. PCx: An interior-point code for linear programming. *Optimization Methods and Software*, Taylor & Francis, v. 11, n. 1-4, p. 397–430, 1999.

DASSIOS, I.; FOUNTOULAKIS, K.; GONDZIO, J. A Preconditioner for A Primal-Dual Newton Conjugate Gradients Method for Compressed Sensing Problems. *SIAM Journal on Scientific Computing*, SIAM, v. 37, n. 6, p. A2783–A2812, 2015.

DOLAN, E. D.; MORÉ, J. J. Benchmarking optimization software with performance profiles. *Mathematical programming*, Springer, v. 91, n. 2, p. 201–213, 2002.

DONOHO, D. L.; HUO, X. Uncertainty principles and ideal atomic decomposition. *Information Theory, IEEE Transactions on*, IEEE, v. 47, n. 7, p. 2845–2862, 2001.

DONOHO, D. L.; TANNER, J. Precise undersampling theorems. *Proceedings of the IEEE*, IEEE, v. 98, n. 6, p. 913–924, 2010.

DUARTE, M. F.; BARANIUK, R. G. Spectral compressive sensing. *Applied and Computational Harmonic Analysis*, Elsevier, v. 35, n. 1, p. 111–129, 2013.

DUARTE, M. F.; DAVENPORT, M. A.; TAKBAR, D.; LASKA, J. N.; SUN, T.; KELLY, K. F.; BARANIUK, R. G. Single-pixel imaging via compressive sampling. *IEEE signal processing magazine*, IEEE, v. 25, n. 2, p. 83–91, 2008.

DUFF, I. S.; MEURANT, G. A. The effect of ordering on preconditioned conjugate gradients. *BIT Numerical Mathematics*, Springer, v. 29, n. 4, p. 635–657, 1989.

ELAD, M.; BRUCKSTEIN, A. M. A generalized uncertainty principle and sparse representation in pairs of bases. *Information Theory, IEEE Transactions on*, IEEE, v. 48, n. 9, p. 2558–2567, 2002.

FIROOZ, M. H.; ROY, S. Network tomography via compressed sensing. In: *Global Telecommunications Conference (GLOBECOM 2010)*. Miami, FL, USA: IEEE, 2010. p. 1–5.

FOUCART, S. A note on guaranteed sparse recovery via l1-minimization. *Applied and Computational Harmonic Analysis*, Elsevier, v. 29, n. 1, p. 97–103, 2010.

FOUCART, S.; RAUHUT, H. *A mathematical introduction to compressive sensing*. New York, NY: Birkhäuser Basel, 2013. (Applied and Numerical Harmonic Analysis).

FOUNTOULAKIS, K. *Higher-Order Methods for Large-Scale Optimization*. Tese (Doutorado) — University of Edinburgh, 2015.

FOUNTOULAKIS, K.; GONDZIO, J. A second-order method for strongly convex l1-regularization problems. *Mathematical Programming*, Springer, p. 1–31, 2013.

GHIDINI, C. T.; OLIVEIRA, A.; SORENSEN, D. Computing a hybrid preconditioner approach to solve the linear systems arising from interior point methods for linear programming using the conjugate gradient method. *Annals of Management Science*, International Center for Business & Management Excellence, v. 3, n. 1, p. 43, 2014.

GOLUB, G. H.; Van Loan, C. F. *Matrix computations*. 3. ed. Baltimore, MD: JHU Press, 2012.

GOLUBOV, B.; EFIMOV, A.; SKVORTSOV, V. *Walsh series and transforms: theory and applications*. Dordrecht, Netherlands: Springer Science & Business Media, 2012. v. 64.

HANDA, A.; NEWCOMBE, R. A.; ANGELI, A.; DAVISON, A. J. Applications of Legendre-Fenchel transformation to computer vision problems. *Department of Computing at Imperial College London, DTR11-7*, v. 45, 2011.

HUANG, G.; JIANG, H.; MATTHEWS, K.; WILFORD, P. Lensless imaging by compressive sensing. In: *20th IEEE International Conference on Image Processing (ICIP)*. Melbourne, Australia: IEEE, 2013. p. 2101–2105.

IVANOV, G. E. Strong and weak convexity for linear differential games. In: *Proceedings of the 35th IEEE Conference on Decision and Control*. Kobe, Japan: IEEE, 1996. v. 4, p. 3729–3734.

IZMAILOV, A.; SOLODOV, M. *Otimização*: Condições de otimalidade, elementos de análise convexa e de dualidade. Rio de Janeiro, RJ: IMPA, 2009. v. 1.

JONES, M. T.; PLASSMANN, P. E. An improved incomplete Cholesky factorization. *ACM Transactions on Mathematical Software (TOMS)*, ACM, v. 21, n. 1, p. 5–17, 1995.

KAKADE, S.; SHALEV-SHWARTZ, S.; TEWARI, A. On the duality of strong convexity and strong smoothness: Learning applications and matrix regularization. *Unpublished Manuscript, http://ttic. uchicago. edu/shai/papers/KakadeShalevTewari09. pdf*, 2009.

KARMARKAR, N. A new polynomial-time algorithm for linear programming. In: *Proceedings of the sixteenth annual ACM symposium on Theory of computing*. Washington D. C.: ACM, 1984. p. 302–311.

KIKUCHI, P. A. *Métodos de pontos interiores aplicados à basis pursuit*. Dissertação (Mestrado) — IMECC-UNICAMP, Campinas, 2013.

LAVERS, C. *Reeds Introductions: Physics Wave Concepts for Marine Engineering Applications*. London: Bloomsbury Publishing, 2017.

LI, C.; YIN, W.; JIANG, H.; ZHANG, Y. An efficient augmented Lagrangian method with applications to total variation minimization. *Computational Optimization and Applications*, Springer, v. 56, n. 3, p. 507–530, 2013.

LUSTIG, M.; DONOHO, D.; PAULY, J. M. Sparse MRI: The application of compressed sensing for rapid MR imaging. *Magnetic resonance in medicine*, Wiley Online Library, v. 58, n. 6, p. 1182–1195, 2007.

MEHROTRA, S. On the implementation of a primal-dual interior point method. *SIAM Journal on optimization*, SIAM, v. 2, n. 4, p. 575–601, 1992.

MEIJERINK, J.; Van Der Vorst, H. A. An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix. *Mathematics of computation*, v. 31, n. 137, p. 148–162, 1977.

MEYER, C. D. *Matrix analysis and applied linear algebra*. Philadelphia, PA: Siam, 2000. v. 2.

MUNKSGAARD, N. Solving sparse symmetric sets of linear equations by preconditioned conjugate gradients. *ACM Transactions on Mathematical Software (TOMS)*, ACM, v. 6, n. 2, p. 206–219, 1980.

NESTEROV, Y. *Introductory lectures on convex optimization*. New York, NY: Springer Science & Business Media, 2004. v. 87.

_____. Smooth minimization of non-smooth functions. *Mathematical programming*, Springer, v. 103, n. 1, p. 127–152, 2005.

NOCEDAL, J.; WRIGHT, S. *Numerical optimization*. 2. ed. New York, NY: Springer Science & Business Media, 2006.

OLIVEIRA, A. R.; SORENSEN, D. C. A new class of preconditioners for large-scale linear systems from interior point methods for linear programming. *Linear Algebra and its applications*, Elsevier, v. 394, p. 1–24, 2005.

RAO, K. R.; YIP, P. *Discrete cosine transform: algorithms, advantages, applications*. San Diego, CA: Academic press, 2014.

ROCKAFELLAR, R. T. *Convex analysis*. Princeton, NJ: Princeton university press, 2015.

RUGGIERO, M. A. G.; LOPES, V. L. d. R. *Cálculo numérico*: aspectos teóricos e computacionais. São Paulo, SP: Makron Books do Brasil, 1997.

SAAD, Y.; Van Der Vorst, H. A. Iterative solution of linear systems in the 20th century. *Journal of Computational and Applied Mathematics*, Elsevier, v. 123, n. 1, p. 1–33, 2000.

SILVA, L. M. da. *Modificações na fatoração controlada de Cholesky para acelerar o precondicionamento de sistemas lineares no contexto de pontos interiores*. Tese (Doutorado) — UNICAMP, 2014.

SIMÕES, L. E. A. *Novos métodos incrementais para otimização convexa não-diferenciável em dois níveis com aplicações em reconstrução de imagens em tomografia por emissão*. Dissertação (Mestrado) — Universidade de São Paulo, 2013.

TREFETHEN, L. N.; BAU III, D. *Numerical linear algebra*. Philadelphia, PA: Siam, 1997. v. 50.

VAITER, S.; PEYRÉ, G.; DOSSAL, C.; FADILI, J. Robust sparse analysis regularization. *Information Theory, IEEE Transactions on*, IEEE, v. 59, n. 4, p. 2001–2016, 2013.

VELAZCO, M.; OLIVEIRA, A. R. d.; CAMPOS, F. A note on hybrid preconditioners for large-scale normal equations arising from interior-point methods. *Optimization Methods & Software*, Taylor & Francis, v. 25, n. 2, p. 321–332, 2010.

_____. Heuristics for implementation of a hybrid preconditioner for interior-point methods. *Pesquisa Operacional*, SciELO Brasil, v. 31, n. 3, p. 579–591, 2011.

WIPF, D.; RAO, B. l0-norm minimization for basis selection. In: *Advances in Neural Information Processing Systems 17*. Cambridge, MA: MIT Press, 2005. p. 1513–1520.

WRIGHT, S. J. *Primal-dual interior-point methods*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 1987. v. 54.

_____. *Primal-dual interior-point methods*. Philadelphia, PA: SIAM, 1997.

ZIA, R. K.; REDISH, E. F.; MCKAY, S. R. Making sense of the Legendre transform. *American Journal of Physics*, American Association of Physics Teachers, v. 77, n. 7, p. 614–622, 2009.

Apêndices

APÊNDICE A

Norma Dual

Seja $\|\cdot\|$ uma norma em $\mathbb{R}^n$. A norma dual associada, denotada por $\|\cdot\|_*$ é definida como

$$\|z\|_* = \sup\left\{z^T x / \ \|x\| \leqslant 1\right\}.$$

De fato, pode-se mostrar que a normal dual satisfaz as propriedades de norma, portanto define uma norma.

A norma dual pode ser interpretada como o operador norma de z^T, sendo z^T uma matriz $1 \times n$, com a norma $\|\cdot\|$ em $\mathbb{R}^n$, e o valor absoluto em $\mathbb{R}$ (a definição de operador norma pode ser encontrada na seção (A.1.5) de Boyd e Vandenberghe (2004)):

$$\|z\|_* = \sup\left\{\left|z^T x\right| / \ \|x\| \leqslant 1\right\}.$$

Podemos obter a seguinte desigualdade da definição de norma dual:

$$z^T x \leqslant \|x\| \, \|z\|_* \, , \text{para todo } x \text{ e } z.$$

Demonstração: Tome y qualquer,

$$z^T \frac{y}{\|y\|} \leqslant \|z\|_* \Rightarrow z^T y \leqslant \|y\| \, \|z\|_* \text{ para todo } y \text{ e } z. \qquad \blacksquare$$

Para qualquer x existe um z para a qual a desigualdade satisfaz a igualdade. Similarmente, para qualquer z existe um x que se obtém na igualdade.

O dual da norma dual é a norma original: $\|x\|_{**} = \|x\|$ para todo x (isto não necessariamente é válido em espaços vetoriais de dimensão infinita).

O dual da norma Euclidiana é a norma Euclidiana:

$$\sup\left\{z^T x / \ \|x\|_2 \leqslant 1\right\} = \|z\|_2 .$$

Demonstração: $\sup\left\{z^T x/\ \|x\|_2 \leqslant 1\right\} = \sup\left\{\left|z^T x\right|/\ \|x\|_2 \leqslant 1\right\}$. Usando a Desigualdade de Cauchy Schwarz,

$$\left|z^T x\right| \leqslant \|z\|_2 \|x\|_2 .$$

Portanto temos:

$$\sup\left\{z^T x/\ \|x\|_2 \leqslant 1\right\} = \sup\left\{\left|z^T x\right|/\ \|x\|_2 \leqslant 1\right\} \leqslant \sup\left\{\|z\|_2 \|x\|_2/\ \|x\|_2 \leqslant 1\right\},$$

como podemos notar, o supremo dar-se-á quando $\|x\|_2 = 1$, obtendo dessa forma a seguinte desigualdade:

$$\sup\left\{z^T x/\ \|x\|_2 \leqslant 1\right\} \leqslant \|z\|_2 .$$

Para $x = \dfrac{z}{\|z\|_2}$, verifica-se a igualdade. Portanto, $\sup\left\{z^T x/\ \|x\|_2 \leqslant 1\right\} = \|z\|_2$. ■

O dual da norma l_∞ é a norma l_1:

$$\sup\left\{z^T x/\ \|x\|_\infty \leqslant 1\right\} = \sum |z_i|_{i=1}^n = \|z\|_1 .$$

Demonstração: O supremo ocorre quando:

$$x_i = \begin{cases} 1, & \text{se } z_i > 0 \\ -1, & \text{se } z_i < 0 \end{cases} . \tag{A.1}$$

Assim $\sup\left\{z^T x/\ \|x\|_\infty \leqslant 1\right\} = \sum_{i=1}^{n} |z_i|$, e o dual da norma l_1 é a norma l_∞ :

$$\sup\left\{z^T x/\ \|x\|_1 \leqslant 1\right\} = \max_i |z_i| = \|z\|_\infty .$$ ■

De forma geral, o dual da norma l_p é a norma l_q, onde q satisfaz $\dfrac{1}{p} + \dfrac{1}{q} = 1$, isto é, $q = \dfrac{p}{(p-1)}$.